INVESTIGATION OF TRANSFER LEARNING FOR ASR USING LF-MMI TRAINED NEURAL NETWORKS

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

[1]Center of Language and Speech Processing
[2]Human Language Technology Center Of Excellence,
Johns Hopkins University, Baltimore, MD

{pghahre1,vmanoha1,hhadian,khudanpur}@jhu.edu, dpovey@gmail.com

ABSTRACT

It is common in applications of ASR to have a large amount of data out-of-domain to the test data and a smaller amount of in-domain data similar to the test data. In this paper, we investigate different ways to utilize this out-of-domain data to improve ASR models based on Lattice-free MMI (LF-MMI). In particular, we experiment with multi-task training using a network with shared hidden layers; and we try various ways of adapting previously trained models to a new domain. Both types of methods are effective in reducing the WER versus in-domain models, with the jointly trained models generally giving more improvement.

Index Terms: Transfer learning, weight transfer, LF-MMI, multi-task learning, Automatic Speech Recognition

1. INTRODUCTION

Transfer learning is the general machine learning approach of transferring knowledge from one model to another model, that can be used in a different, but related task or domain and it can be regarded as a superset of unsupervised adaptation, domain adaptation, model compression and many other related problems. There is a rich survey of transfer learning methods in the literature [1, 2, 3]. In this work, we investigate the use of transfer learning in automatic speech recognition (ASR) tasks for adapting neural networks to different domains or datasets

One of the advantages of deep learning is to particularly learn a hierarchy of feature representations from low-level features to more abstract higher-level features [4, 3], consequently it can be useful in transfer learning. Multi-task learning [5] has been adopted to explicitly learn intermediate-level features in the neural network that are useful for several different tasks. In an alternate paradigm, pretraining [6, 7] has been used to implicitly learn intermediate representations that are useful for different tasks. The intermediate layers in neural networks trained on speech data appear to be not specific to any particular task, while the higher layers are task-specific [8]. This has been demonstrated in [8], where unsupervised pretraining using Deep Belief Networks (DBN) has been shown to learn representations useful for phoneme recognition and audio classification tasks. Unsupervised pretraining has also been applied to multilingual speech recognition [9]. Supervised training using out-of-domain data is also a form of pretraining and it has been used to learn multilingual bottleneck features in [10, 11].

Transfer learning methods have been applied to speech processing in various settings. Wang et. al [12] gives a good overall survey of methods used in speech processing. Domain adaptation by adapting network parameters, and in particular speaker adaptation, has been attempted using simple Linear Input Network (LIN) [13]. This has inspired more advanced methods like fDLR [14] and linear transforms at various stages of the network [15] using Liner Hidden Network (LHN). The weight transfer method described in this work is similar to LHN-based adaptation, but we re-initialize an affine layer instead of training a newly added layer. LHN-based adaptation is compared with multi-task learning in [16]. A speaker adaptive training (SAT) type approach is investigated for speaker adaptation of DNN by learning hidden unit contributions (LHUC) [17]. In [18], several transfer learning approaches for speaker adaptation are compared including multi-task learning. Multi-task architectures with hidden layers shared across languages have been used successfully for multilingual training [19, 20]. Not only the amount of data, but also the similarity of the languages i.e. the relatedness of the task is found to be important for effective transfer learning [21, 22].

In this work, we investigate 2 different approaches to transfer knowledge between networks. For this work, we circumvent the side-effect of language similarity (or dissimilarity) seen in multilingual training and focus only on English datasets, albeit in different language domains and environments (channels). We find that generally multi-task training performs better than weight transfer. However weight transfer is still effective compared to the unadapted model and might be preferable as it does not require training on both source and target data as in the multi-task training approach. We investi-

gate weight transfer approaches and show that a single-stage training of transferred layers in weight training is better than a two-stage one that includes a final fine-tuning. We also find that even a weak model can be a good seed model for transfer learning, and thus we do not need to train the seed model to convergence on the source data. We also apply transfer learning across datasets with different sampling rates. Contrary to the popular approach of down-sampling data, we show that up-sampling the data is the better approach. We investigate the effect of i-vector mismatch across domains and conclude that it is best to train the i-vector on the combined source and target data. We investigate whether the transfer learning approaches are as applicable to sequential objectives as they are frame-level cross-entropy objective, and conclude that they are.

This paper is organized as follows.Section 2 discusses the transfer learning approaches investigated in this paper; multi-task learning 2.1 and weight transfer 2.2 . In section 3, we investigate weight transfer approach as a function of number of transferred layers from source model, performance of source model on source domain, amount of target data and different mismatch conditions. Section 3.3 analyzes the effectiveness of two proposed approaches and section 3.4 studies the effect of sequence level versus frame-level objectives in transferring knowledge across datasets using two methods.

2. MODEL DESCRIPTION

In this section, we investigate 2 different approaches to transfer knowledge between data sets for automatic speech recognition. Section 2.1 describes joint multi-task approach for transfer learning and Section 2.2 describes the weight transfer approach.

2.1. Joint multi-task learning

In this approach, we used the setup where the initial layers of the network are shared across all tasks and each task has a specific final layer. This approach has been previously used in the several works including [5, 20, 19]. If tasks are known to have different importance, then they can be weighted proportionally as in [22]. Unlike [22], which uses model averaging (typically after training over 400000 frames), we train for different tasks in different mini-batches, which averages over a minibatch (typically 10000 frames). Training the network this can reduce optimization difficulty due to co-adaptation. Another issue is over-training to a specific task, which might degrade performance in other tasks as seen in [21] when transferring from Fisher English to other languages. To reduce such over-training effect, the gradients are scaled for each task by a factor inversely proportional to the square root of the number of training samples in that task.

2.2. Weight Transfer

The main idea here is that the internal layers of DNN learn intermediate-level representations of input, which can be pre-trained on one dataset (or task) and re-used on the other tasks. A typical weight transfer approach is to first train the model on a large dataset, retain only n layers and add new task-specific adaptation layers over those.

The usual strategy is to do a two-stage training by freezing the transferred layers and train task-specific layers in the 1^{st} training stage and then fine-tune the whole network in the 2^{nd} stage of training using a smaller learning rate [23]. However, we show in Section 3.1.1 that it is better to do a single-stage training – train the transferred layers with a smaller learning rate while training the task-specific layers with a larger learning rate.

3. RESULTS AND DISCUSSION

In this section, we discuss transfer learning experiments to investigate the effectiveness of our two transfer learning methods – weight transfer and multitask learning – under various source and target conditions. For the experiments, we use time-delay neural networks [24] (TDNN) with i-vectors [25] for speaker adaptation [26]. For the details about the training of TDNN with lattice-free maximum mutual information (LF-MMI) objective, the reader is directed to [27]. As in that work, we train the network with LF-MMI objective and cross-entropy regularization. We use several different corpora for our experiments – Switchboard (SWBD), Librispeech [28], WSJ and AMI [29] in both individual headset microphone (IHM) and single distance microphone (SDM) conditions.

This section is organized as follows. In section 3.1, various training strategies and amount of transferred layers in weight transfer approach are discussed. Moreover the effectiveness of the weight transfer approach is investigated as function of the amount of data in the source vs target domains and the performance of source model on the source domain. In section 3.2, we discuss how transfer learning is affected by various mismatch conditions. In section 3.3, we compare the weight transfer method and the multi-task learning method for transfer learning. In section 3.4, we answer the question whether the transfer learning gives more improvement when using frame-level or sequence-level objectives.

3.1. Weight Transfer method

3.1.1. Single-stage vs two-stage training

Table 1 shows results using two different weight transfer strategies. In these experiments, 5 layers of the source model trained on Switchboard dataset are transferred to AMI-SDM dataset and 2 randomly initialized layers are added on top of transferred layers. The global learning rate is the same in all stages of experiments and the learning rate for each layer is

Table 1. *single-stage vs. two-stage WER Results on SWBD→AMI-SDM.*

#	Model	LR factors α	LR factors β	# epochs s_1	# epochs s_2	WER% dev	WER% eval
1	Baseline	1	-	4		45.3	50.0
2	two-stage (s1)	0.25	1	4	2	50.6	55.0
	two-stage (s2)					46.5	51.2
3	two-stage (s1)	0.25	1	2	2	51.8	56.3
	two-stage (s2)					46.4	51.5
4	single-stage	0.02	-	4	-	45.4	50.3
5	single-stage	0.1	-	4	-	44.5	49.7
6	single-stage	0.1	-	2	1	44.5	49.4
	single-stage*					44.3	49.5
7	single-stage	0.25	-	2	-	44.0	48.9

*: fine-tune whole net

the global learning-rate scaled by its learning rate factor. In two-stage training, the transferred layers are fixed and only the task-specific layers are trained in the first stage with a learning rate factor α for s_1 epochs. Then, in the second stage, the whole network is fine tuned using a smaller learning rate factor β for s_2 epochs. In single stage training, the transferred layers are trained with a learning rate factor α, while newly added layers are trained with the global learning rate, all for s_1 epochs.

As shown in the table, single-stage training gives better results than the conventional two-stage training with a smaller number of epochs. The single-stage results improve as we increase the learning rate factor α.

In addition, fine-tuning the single-stage trained model by training whole model with smaller learning rate, does not improve the results as shown in experiment 6 in table 1. In single-stage* model, the single-stage trained model is fine-tuned for $s_2 = 1$ epoch.

3.1.2. *Effect of number of transferred layers*

The initial layers in the deep neural networks are "generic" and final layers are task-specific; so there must be a transition boundary from generic to specific in some layer. To investigate this, we conduct two weight transfer experiments to target corpus AMI in IHM condition, one with Librispeech as the source corpus and the other with Switchboard as the source corpus. The number of layers in the Librispeech and SWBD neural networks were 6 and 7 respectively. The neural networks all had TDNN architectures with the same overall input context.

In the first case with Librispeech as the source, the results in Figure 1 show that the largest WER reduction was achieved by transferring half of the layers (3 or 4 layers out of 7). On the other hand, for the case of Switchboard as the source corpus, the largest WER reduction was achieved by transferring a larger proportion of layers (5 layers out of 6).

The reason for a larger proportion of transferred layers being better in the case of SWBD might be that SWBD and AMI-IHM senones are more similar compared to Librispeech and AMI-IHM. This might be expected because Switchboard and AMI-IHM are both spontaneous speech corpora, while Librispeech is a read speech corpus.

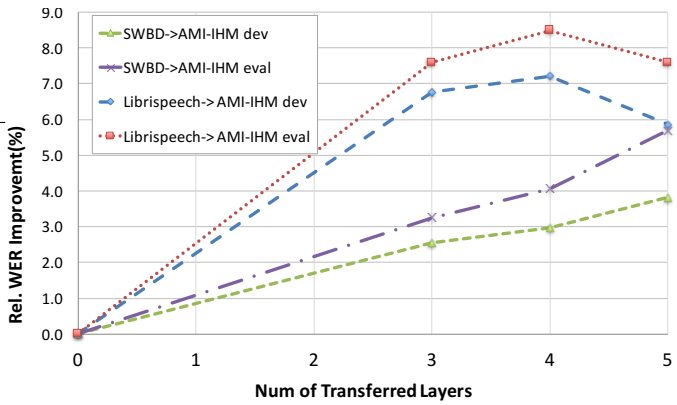

Fig. 1. *WER(%) vs Number of transferred layers for Switchboard to AMI*

In the weight transfer approach described above, the last layer is not usually transfered due to the fact that the phone set and the tree in the source and the target domain are different. However, in some cases we can use the same phone set for both the source and the target data, and hence share the tree and the senones. In these cases, we can transfer the whole network, including the last layer. This is particularly useful in cases where the target corpus is very small compared to the source corpus, as found in the case of MGB-3 challenge [30].

We share the phone sets for Librispeech and WSJ and do weight transfer from Librispeech to WSJ by transferring all the layers. Figure 2 shows the results of weight transfer for different number of transferred layers including the whole network transfer of 7 layers. Here transferring all the layers gives the best WER performance.

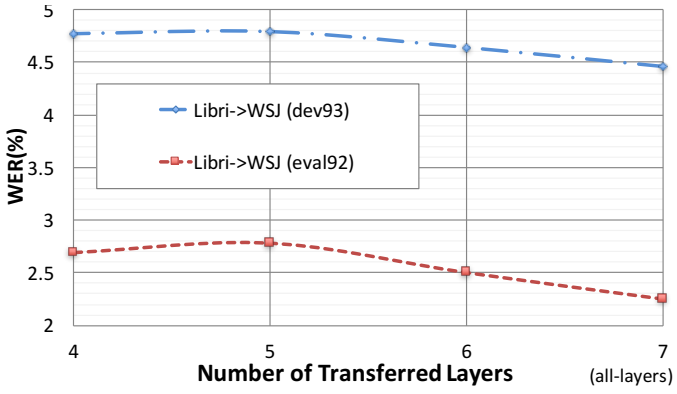

Fig. 2. WER(%) vs Number of transferred layers for Librispeech to WSJ

3.1.3. *Effect of amount of target data*

To investigate the effect of the amount of data in the target corpus in transfer learning, we conduct an experiment with transfer learning from 1000 hours Librispeech corpus to 80 hours WSJ. The amount of data in the target WSJ corpus is varied by using subsets containing 84 (15h), 144 (40h) and 284 (80h) speakers respectively. For comparison, the results on training directly on WSJ data subsets i.e. without transfer learning, are shown as "Baseline". In all these experiments, the i-vector extractor is trained on only Librispeech data.

Figure 3 shows the WER results for Baseline WSJ and transferred model trained using the WSJ subsets. As seen in the figure, the improvement using transfer learning is the largest with 84 speakers and it reduces as we add more data to the target corpus. The results show that weight transfer is most effective when the amount of data in the target corpus is small and insufficient to train a good model. Interestingly, we can see that the improvement to the 84-speaker WSJ model due to weight transfer from the Librispeech model trained on another domain is more than the improvement gained by using the rest of 200 in-domain speakers in WSJ data (i.e. extra 65 hrs in-domain data).

As explained before, the phone sets and lexicons in Librispeech and WSJ are similar which allows us to transfer the final layer too. Therefore, we tried whole network transfer with varying amount of target data. Since the final layer in DNNs is usually a large transformation with dimensionality of hidden layer size by number of senones, training this layer from scratch can be more difficult when there is less amount of training data in the target domain. As a result we expect transferring already-trained final layer to be more helpful in such cases. This can be observed in figure 3.

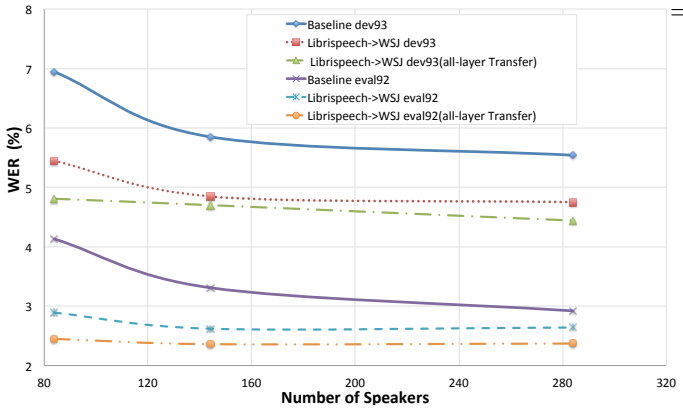

Fig. 3. *WER(%) vs size of target WSJ corpus (in number of speakers) for baseline and transferred model from Librispeech*

3.1.4. *Effect of power of source model*

In this section, we investigate the effectiveness of weight transfer method as a function of the number of parameters used in source model and the number of epochs used to train it. We answer whether a weak source model (as measured on source test set performance) can still be useful as a seed for weight transfer. Table 2 shows the WER results for weight transfer from SWBD to AMI in both IHM and SDM conditions. The baseline source model in SWBD was trained for 4 epochs. The first column of the table shows WER results on *eval2000* test set for Switchboard using different source model. The results using the baseline model for weight transfer is shown in row 1 of the table. If instead, we train a model that has only 30% the number of parameters as the baseline model, the performance on *eval2000* test set drops by 0.8% to 17.8%. Using this as seed for weight transfer, we get a WER performance that is clearly worse as shown in row 2. However, if we train the same model as in the baseline, but for fewer epochs like 1 or 2 instead of 4 epochs, and use it as seed for weight transfer, the WER performance is closer to that of the fully-trained baseline model. This is in spite of the seed model having a WER (17.9%) on the source domain as poor as the model that has 30% number of parameters (17.8%).

These results suggest that the source model can learn some generic features from the source data in the initial stages of training that are useful for the target data. But the later stages of training learn features specific to the source data that are not very useful for the target data.

Table 2. *WER(%) results for different source models: SWBD → AMI.*

	Source	Target corpus			
	SWBD	AMI-SDM		AMI-IHM	
Model	eval2000	dev	eval	dev	eval
4 epochs	17.0	43.5	48.7	22.5	22.8
4 epochs 30% Params	17.8	45.6	50.3	23.3	23.6
1 epoch	17.9	43.8	48.9	22.8	23.3
2 epochs	17.1	43.5	48.8	22.6	23.1

3.2. Transfer learning in mismatch conditions

3.2.1. *Transfer learning in sampling rate mismatch condition*

One of the main difficulties in transfer learning for ASR is that of sampling rate mismatch of the recordings (e.g. 8kHz, 16kHz, etc). Transferring information across datasets with different sampling rate requires down-sampling of data, which results in some loss of high-frequency information and degrades ASR performance. To verify this, we trained two separate TDNNs – one using MFCC features extracted from 16kHz data and the other from down-sampled 8kHz data–

on AMI corpus in SDM condition. The WER results are in the first two rows of table 3, which show that removing high-frequency information results in 3 to 4% WER degradation.

We used the transfer learning approach, to adapt the 8kHz trained network to the 16kHz features on AMI-SDM dataset. Here, we retrain the first affine transform after input features with some learning rate and fine-tune the rest of the network with much smaller learning rate. In the experiment indicated by $(*)$, a new affine transform is added before LDA layer of the 8kHz trained network and this transform is initialized to regress the 16kHz features to 8kHz features. The results are in rows 3 and 4 of table 3. They show that that the information loss due to pretraining on down-sampled data as opposed to the full-band 16kHz data can only be partially recovered using a simple learning new adaptation layer at the initial layer.

Table 3. *WER (%) results on AMI-SDM*

Model	WER	
	dev	eval
16kHz	40.5	44.4
8kHz	43.6	48.6
8 ⇒16 kHz Transfer	41.9	46.2
8 ⇒16 kHz Transfer*	42.4	47.2
16kHz SWBD⇒AMI	39.4	43.9

Mixed-bandwidth ASR training, which combines narrow-band (inserting zeros for high-frequency bands) and wide-band speech signal can improve ASR performance on narrow-band test domain [31]. The second transfer learning approach is to train the source model using MFCC features extracted from up-sampled 16kHz data and transfer the layers from up-sampled source model using the weight transfer approach in section 2.2. We tried this by using a source model trained on up-sampled 16kHz Switchboard data as seed model for weight transfer to AMI-SDM. As shown in the row 4 of table 3, this approach gives 1% absolute WER improvement over the baseline that uses only 16kHz AMI-SDM data.

3.2.2. Effect of i-vector extractor

We use i-vector based speaker adaptation of neural networks. In the weight transfer learning approach, we have to use the same i-vector extractor for both training on source data as well as adaptation to target data. In this section, we investigate the effect of using different i-vector extractors. We conduct weight transfer experiments from Switchboard to AMI in SDM condition (down-sampled to 8kHz) using i-vector extractors trained in 3 different ways – one trained only on the source (SWBD) data, one trained on 25% subset of data from source and target, and one trained only on the target (AMI) data. The "Baseline" columns of table 4 reports results on directly training the neural network on the AMI data i.e. without transfer learning. Comparing the last three rows with the first row showing WER without i-vector adaptation, we can see that any of the 3 extractors gives $3-4\%$ absolute improvement. This suggests that even an out-of-domain i-vector extractor is suitable for speaker adaptation in ASR. These are also 1-2% better than CMVN normalization results shown in row 2 of the table. The "Weight transfer" columns in table 4 shows that using the extractor trained on combined data (row 4) gives more than 1% absolute improvement over using i-vector extractor trained only on the source data (row 3).

Table 4. *Speaker adaptation:* 8kHz *SWBD* → 8kHz *AMI-SDM WER(%) results*

Extractor	Baseline		Weight Transfer	
	dev	eval	dev	eval
No adaptation	49.5	53.7	45.7	50.0
CMVN adaptation	48.0	52.7	45.8	50.7
iVector adaptation				
SWBD extractor	46.2	51.1	44.6	49.5
Combined extractor	45.8	50.8	43.5	48.7
AMI-SDM extractor	45.3	50	-	-

3.2.3. Transfer learning in environment mismatch

In this section, we discuss experiments showing transfer learning from Librispeech to AMI in IHM and SDM conditions. The results are in table 5. The weight transfer model gives 1% absolute improvement in WER in the case of SDM condition, and 2% absolute improvement in WER in the case of IHM condition. This might suggest that having the source and target data from a similar environment condition (like Librispeech and AMI IHM) is better for the weight transfer scenario.

Table 5. *WER results: Librispeech to AMI Transfer.*

Target Data	System	WER(%) dev	eval
AMI-SDM	Baseline	41.0	45.2
	Weight Transfer	39.9	44.2
AMI-IHM	Baseline	22.2	22.4
	Weight Transfer	20.6	20.5

3.3. Weight transfer vs Multi-task training

In this section, two transfer learning approaches are investigated for transferring information from 300 hrs Switchboard dataset to AMI-IHM, AMI-SDM and WSJ datasets. As discussed in Section 3.2.1, down-sampling the data degrades performance on AMI dataset. So we report results on the 8kHz baseline for all the corpora. The results in Table 6 show good improvement over baseline using both weight transfer and multi-task training.

We also tried a multi-task approach, where data is pooled from all 3 target datasets and all layers except last layer are shared across all datasets. This is reported in the Multi-task-pool row and shows slight improvement over Multi-task using just source and target data in the cases of AMI-SDM and AMI-IHM.

Table 6. *WER results: SWBD to AMI and WSJ Transfer.*

	WER		Rel. WER(%)	
AMI-SDM	*dev*	*eval*	*dev*	*eval*
Baseline	45.3	50	-	-
Weight Transfer	43.9	49.3	3.1	1.4
Multi-task	45	49.2	0.66	1.6
Multi-task-pool	44.9	49.6	0.9	0.8
AMI-IHM	*dev*	*eval*	*dev*	*eval*
Baseline	23.6	24.6	-	-
Weight Transfer	22.7	23.2	3.8	5.7
Multi-task	22.4	22.7	5.1	7.7
Multi-task-pool	22.1	22.6	6.4	8.2
WSJ	*dev93*	*eval92*	*dev93*	*eval92*
Baseline	5.49	3.15	-	-
Weight Transfer	5.32	2.84	3.1	9.8
Multi-task	4.8	2.57	12.5	18.5
Multi-task-pool	4.99	2.53	9.1	19.7

Multi-task-pool: Trained on pooled speed-perturbed SWBD, AMI-SDM, AMI-IHM, WSJ datasets.

3.4. Transfer learning using different objectives

The state-of-the-art neural networks in ASR are trained with sequence-level objectives like LF-MMI [27]. Frame-level objectives used in model transfer such as using soft-targets [32, 33] are not naively applicable to LF-MMI objective as the neural network outputs are not frame-level posteriors. Regressing information too close to the output may not be applicable as the outputs are not specifically trained for good frame-wise predictions. Furthermore, output nodes in the LF-MMI networks operate at a lower (one-third) frame rate.

Table 7 shows transfer learning results using frame-level cross entropy versus sequence-level LF-MMI objective. The i-vector extractor is trained on 25% of the combined data from all the datasets for all the experiments, and all datasets are down-sampled to $8kHz$. In the multi-task training experiments, the TDNN models are trained on pooled speed-perturbed datasets of Switchboard, AMI-SDM, AMI-IHM and WSJ using cross-entropy and LF-MMI objectives. In weight transfer experiments, both source and target models are trained using same objective function i.e. both Cross-entropy or both LF-MMI. Switchboard is used as the source dataset for the weight transfer experiments. The results show that transfer learning is as effective for LF-MMI objective as is for frame-level cross-entropy objective.

Table 7. *Transfer Learning for frame-level CE vs. sequence-level LF-MMI objective*

	Cross-Entropy		*LF-MMI*	
WSJ	*dev93*	*eval92*	*dev93*	*eval92*
Baseline	6.38	3.38	5.49	3.15
Multi-task*	5.85	3.47	4.99	2.53
AMI-IHM	*dev*	*eval*	*dev*	*eval*
Baseline	26	27.6	23.6	24.6
Multi-task*	23.7	25.1	22.1	22.6
Weight Transfer	25.2	26.3	22.7	23.2

*: Trained on pooled speed-perturbed SWBD, AMI-SDM, AMI-IHM, WSJ datasets.

4. CONCLUSIONS

We have investigated two transfer learning approaches – weight transfer and multi-task training – in ASR using sequence-trained neural network based on Lattice-free MMI in different acoustic conditions. We present results on different small-sized LVCSR tasks with 80-100 hours of data by transferring knowledge from larger corpora with 300-1000 hours. Generally we found that multi-task training performs better than weight transfer. However weight transfer is still effective compared to the unadapted model, and hence it might be preferable over multi-task training as it does not require re-training on the pooled data. The results for weight transfer show that single-stage training of transferred layers with very small learning rate, while training target-specific layers is better than 2-stage training by freezing the transfered layers at the 1^{st} stage and fine-tuning the whole network at 2^{nd} stage. Our experiments show that even a model trained on source data for half or quarter the number of epochs is as effective a seed model for weight transfer as a fully-trained model. We found from our experiments on SWBD and AMI that the most effective way of dealing with sampling rate mismatch across datasets used for transfer learning is to up-sample the data. It was best to train the i-vector extractor on the combined source and target data, although even an i-vector extractor trained only on the out-of-domain source data was quite effective. We finally found that transfer learning is equally applicable to sequence-level objectives like LF-MMI as it is to frame-level cross entropy objective.

5. ACKNOWLEDGEMENTS

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128 and IARPA Contract No 2012-12050800010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

6. REFERENCES

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

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

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

[4] Y. Bengio, F. Bastien, A. Bergeron, N. Boulanger-Lewandowski, T. M. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache *et al.*, "Deep learners benefit more from out-of-distribution examples." in *AISTATS*, 2011, pp. 164–172.

[5] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[6] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[7] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[8] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[9] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in dnn-based lvcsr," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 246–251.

[10] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. ICASSP*, May 2013, pp. 6704–6708.

[11] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 336–341.

[12] D. Wang and T. F. 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.

[13] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid hmm-ann continuous speech recognition system," 1995.

[14] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 366–369.

[15] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[16] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Rapid adaptation for deep neural networks through multi-task learning," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[18] Z. Huang, S. M. Siniscalchi, and C.-H. Lee, "A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition," *Neurocomputing*, vol. 218, pp. 448–459, 2016.

[19] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 8619–8623.

[20] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*. IEEE, 2013, pp. 7304–7308.

[21] F. Grézl, E. Egorova, and M. Karafiát, "Study of large data resources for multilingual training and system porting," *Procedia Computer Science*, vol. 81, pp. 15–22, 2016.

[22] R. Sahraeian and D. Van Compernolle, "Using weighted model averaging in distributed multilingual dnns to improve low resource asr," *Procedia Computer Science*, vol. 81, pp. 152–158, 2016.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[24] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. 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.

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

[26] M. Karafiat, L. Burget, P. Matejka, O. Glembek, and J. Cernocky, in *Proc. ASRU*. IEEE, Dec., pp. 152–157.

[27] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," 2016.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[29] I. McCowan, J. 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*, vol. 88, 2005.

[30] V. Manohar, D. Povey, and S. Khudanpur, "JHU Kaldi System for Arabic MGB-3 ASR Challenge using Diarization, Audio-Transcript alignment and Transfer learning," in *Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on*, 2017.

[31] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[32] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[33] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.