Adaptation Strategies for Human-Machine Interactions in Dialect Specific Environment

György Szaszák and Piero Pierucci

Telepathy Labs GmBH, Militärstrasse. 36, 8004 Zürich Switzerland
E-mail: gyorgy.szaszak@telepathy.ai, piero.pierucci@telepathy.ai

Abstract: Offering human machine interfaces capable of handling cultural variation in speech is an exciting topic in cognitive info-communication. From a machine learning point-of-view, this paper examines Automatic Speech Recognition (ASR) with respect to transcribing interactions occurring in a language environment where a particular dialect of a pluricentric language is spoken. Conservative retraining, transfer learning, multi-task training, matrix factorization, i-vector based techniques as well as adversarial and teacher-student training are all considered suitable approaches for the adaptation of deep neural acoustic models in ASR. Facing a problem of adaptation for accented speech, which method should be preferred? Comparing these techniques is often complicated, as different experiments are carried out on diverse datasets and within various frameworks. It is also worthwhile analysing possible combination of such techniques within complex systems. The objective of this work is to systematically compare and analyse a number of domain adaptation techniques for ASR using the same framework, in order to allow for a fair comparison on adapting US English acoustic models for particular accents. Our results indicate that, when properly hyperparametrized and carefully regularized, the easiest approaches, requiring less complexity and reduced computational power, can perform equally well as the more complex ones.

Keywords: human-machine interaction; speech recognition; adaptation; dialect; conservative training

1 Introduction

In human-machine infocommunication [1, 2] Automatic Speech Recognition Systems (ASR) are often used to convert speech into text. Just like in most machine learning tasks, where data scarcity is not a hindering factor, the deep learning paradigm has infused ASR technology in the past 5-6 years, leading to formerly unseen low error rates, through neural hybrid or all neural (end-to-end) models [6, 7]. However, when shifting away from the former state-of-the-art Gaussian Mixture Model (GMM) approach in acoustic modelling, some capabilities and elegant engineering solutions [4, 5] were lost, which impact

personalization and social convenience of ASR in a negative way. More precisely, with GMM based acoustic models, adaptation of the acoustic models was a very easy and effective task [3, 4], which moreover, did not require much data in case of Maximum Likelihood Linear Regression (MLLR) based model refinement. The MLLR adaptation made it possible to customize ASR for interactions taking place in specific social and language environments, therefore covering various speech styles and language varieties spoken in different communities.

With the neural network approach in acoustic modelling, no simple adaptation method is available, as the parameters of the acoustic model do not reflect direct spectral (or cepstral) phoneme mapping any more. This is a problem since variation in pronunciation [3, 4, 5, 8, 9] within the same language constitutes a so-called domain mismatch as the acoustic model cannot cope well with data points which are different from the ones seen during the training. Given that, ASR for interactions occurring in an unfamiliar social or language environment lead to a considerable increase in Word Error Rates (WER). This motivated starting up a number of research activities targeting acoustic adaptation of neural acoustic models, which resulted in a multitude of techniques being proposed one after the other. In the following section, we overview some of these, but we also point out that we observe a general tendency for these techniques to became more and more complex, making questionable whether the obtained small performance improvement is worth the large effort required by the adaptation approach. Another problem consists in the large variety of experimental conditions, which makes a comparison close to impossible when various datasets are used for training and benchmarking, where often this experimental condition variety actually reflects different levels of mismatches related to socio-cultural environment.

Unlike GMM acoustic models, which can be adapted by using some minutes of data, neural hybrid acoustic models usually require much more adaptation data to obtain a measurable performance improvement. This is understandable if one considers the relative data scarcity versus the number of parameters of the acoustic model. Training or retraining with low amount of data can lead to catastrophic forgetting or overfitting [10, 11], which means that the model may become corrupted and therefore incapable of generalizing in order to handle unseen cases. To avoid this, some adaptation approaches can apply a slightly different logic to minimize the risks of overfitting: either selecting a subset of the parameters to be updated by freezing all other network parameters (as part of a very conservative retraining with high emphasis on low learning rates) or adopting a more impactful, “aggressive” regularization.

As acoustic model adaptation is unavoidable for model customization for language varieties spoken in different socio-cultural entities, in this paper we are going to review and evaluate several adaptation techniques for ASR with neural acoustic model [12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

Those entities use the same language in human-machine communication, but with slightly different acoustic and linguistic configurations. Our goal is to compare a number of adaptation paradigms within the same framework for the same task to get a clearer picture of their capabilities. As we already discussed, the multitude of baselines and benchmarking datasets makes this comparison close to impossible, despite the rich scientific literature on the acoustic model adaptation subject. Our approach will also allow us to analyse how specific paradigms and training techniques interact – especially those having a regularizing effect and looking at the available amount of adaptation data, both crucial for adaptation and whether the benefits they yield are additive to some extent (that is whether their combination leads to more improvement). We will use a standard English ASR acoustic model trained with the Kaldi toolkit [22] and adapt it for Indian English to demonstrate the results. We also compare adaptation for varieties being less different from the US baseline, including the UK, Canadian, Australian and specific US accents. Please note that the Indian accent itself is not uniform and could be further divided based on geographical consideration, but our datasets did not allow for such a fine grade evaluation in this work.

2 Adaptation Paradigms

Considering adaptation of neural acoustic models, ideas were initially driven by trying to adopt the Maximum Likelihood Linear Regression (MLLR) adaptation framework to non-neural Gaussian Mixture Model (GMM) based acoustic models for ASR. Imposing similar linear or non-linear transformations on one or more network layers [12, 13] was proposed in the so-called Learning of Hidden Unit Contributions (LHUC) model [14]. When LHUC transformation is applied to bottom-wise layers of the neural model, a normalizing effect is produced which makes the model domain invariant, as bottom-wise layers are thought to perform mostly the feature extraction part of the processing. If we use LHUC in the upper-wise (topmost) layers prior to the softmax, we adapt the model in the classical sense, as top layers act as classifiers [15] especially in networks with convolutional alike bottom layers and feed-forward (or eventually recurrent) top layers such as Time Delay Neural Networks (TDNN) [23].

A similar reasoning leads to another adaptation technique based on transfer learning [15], optionally coupled with multi-task training. The idea here is to transfer a feature extractor trained on a large out-of-domain dataset, and train only the classifier from scratch on an in-domain dataset. This approach, referred to as transfer learning hereafter, reduces the number of parameters to be trained and hence allows for using less data keeping the risk of overfitting under control. As basic speech features tend to be less language dependent, usually the bottom layers are transferred, and the top layers are removed or re-initialized (trained

from scratch). Multi-task training [16] is a technique often used in such transfer learning setups, as it allows for an intrinsic normalization of the features, producing less domain dependent bottom layers. A drawback of multitask training is that it takes longer to train, but the most disturbing constraint is that the in-domain adaptation data is required already for the training, which moreover has to be abundant enough to avoid heavy imbalance between the datasets associated with the multiple tasks.

Another popular method exploits matrix factorization from linear algebra in order to adapt some neural model's weight matrices. Factorizing the weight matrix can be regarded as equivalent to adding a linear bottleneck layer between the two specific layers [24]. As the dimension of this bottleneck layer can be configured in a wide range, adaptation is highly scalable and hence allows to adjust the number of parameters to be adapted to the amount of the available data. Some works use factorized neural models by default for acoustic modelling, which has been shown to slightly improve overall performance in [24]. Another often used factorization approach is called Low Rank Plus Diagonal (LRPD) decomposition [17].

Embedding alike vectors, better known as i-vectors, x-vectors [18, 25] or d-vectors [19], can also help in acoustic model adaptation. Hereafter we will use the term i-vector, but thereby referring to other similar embedding vectors such as x- or d-vectors. We can demonstrate their usage if we think like adding a bias in the first hidden layer of the network through the i-vectors [26]. Even if they are tipically used to describe speaker characteristics, they can also be used to capture domain specific variation. Please note that once i-vectors are used in the acoustic model (which is often the case), these intrinsically may already capture part of the variation resulting from domain variance. Eventually, i-vectors can be piped into lightweight auxiliary networks which perform an attention alike rescaling or add bias to the respective parameters in the acoustic model network [17], similar to the LHUC paradigm [14].

Adversarial learning can also be considered for training domain invariant models [20]. Its advantage that it supports unsupervised adaptation. However, as high amount of data is required for various accents during training, it should be regarded as a training method for obtaining invariant models, but the framework cannot be applied to classical adaptation scenarios with small amount of in-domain data. In [27] the authors claim that within the Kaldi ASR toolkit, no benefit was observed in terms of ASR performance when considering adversarial training techniques. The reason for this may be that such methods [20] may be beneficial with very high amount of training data.

Last, but not least, the teacher-student paradigm has also been demonstrated to be leveraged by the Kaldi toolkit for acoustic model training in [21]. Despite being an interesting approach, teacher-student learning requires a parallel corpus for in domain and out of domain data, which is a criterion hard to fulfil in practice, as far as ASR is considered.

3 Material and Methods

3.1 Baseline

For this work we have chosen widely used open-source tools and open-source data to allow for reproducibility of the results. The ASR was trained with the Kaldi toolkit [22], following the Librispeech [28] TDNN [24] standard `1c' recipe on 960 hours of `cleaned' data. For decoding, we simply adopted the open source `tgsmall' grammar, associated with Librispeech data and widely used for benchmarking ASR trained on the Librispeech recipe. We did not consider the adaptation of the language models, but focused exclusively on the acoustic models, which we think have to cope with higher variance in case dialect adaptation is required for human-ASR interactions.

Adding i-vectors to the input was part of our baseline, and it is also part of the default Kaldi Librispeech recipe. This means that part of the variation related to the accent mismatch may therefore be captured by i-vectors, even if the i-vector extractor is trained on only source domain (e.g. Librispeech) data.

3.2 Accented Data

We use the Mozilla Common Voice speech corpus [29] for providing adaptation data in the target domain. In the Common Voice corpus for English, 15 varieties of English are represented, out of which the variants labelled as Australian, Canadian, British (England) and US datasets contain at least 8 hours of good quality transcribed data. For each of these 5 varieties, we created c.a. 1.5 hours test sets to be discarded from training for adaptation.

3.3 Regularization

Regularization and conservative training are very important when performing model adaptation, as we usually work with a low amount of in-domain data, and by typically at least two orders of magnitude difference in dataset sizes (i.e. adapting a baseline trained on 1000 hours with 10 hours of in-domain data). Whereas regularization usually operates by controlling the cost function and preventing sudden changes, the learning rate can also be directly reduced to ensure new data does not modify model parameters in an excessive manner. The learning rate can be specified to be different for each layer and each epoch. This means that bottom most layers can be updated extremely carefully whereas letting topmost layers learn by a slightly higher rate. This coincides with the assumption that bottom layers tend to perform feature extraction, and classification is obtained in the top layers. Reducing the learning rate as iterations follow each other is also an effective and easy way of keeping the training conservative.

A very often used regularization technique in the field of model adaptation is the Kullback-Leibler Divergence (KLD) based regularization [30]. One of its advantages is that KLD can be carried out without any modification in the core training algorithms, but simply by rescaling the targets themselves [30], provided that the objective allows for this (i.e., with an objective based on frame based cross entropy). To achieve this with a cross-entropy minimization objective (or maximizing negative cross entropy), a new target probability distribution has to be created which can be obtained as a linear combination of the original target distribution $\hat{p}$ and the distribution computed via forced alignment with the adaptation data p. So in the objective

$$D_{KL} = \frac{1}{N}\sum_{t=0}^{N-1}\sum_{y\in C}\hat{p}(y|x_t)\log\left(p(y|x_t)\right) \tag{1}$$

dealing with N samples and C senones for the estimated distribution p at time frame t, the target distribution $\hat{p}$ is computed from the original target $\hat{p}_0$ and the estimate of the source model:

$$\hat{p}(y|x_t) = (1-\alpha)\hat{p_0}(y \mid x_t) + \alpha p(y \mid x_t), \tag{2}$$

where α is the regularization coefficient. The term senone may need a brief explanation: a senone is an abstract entity to be modelled that can be linked to the tied state of a traditional triphone model, i.e., the leaves of the decision tree used to represent context in ASR acoustic models.

However, in recent acoustic model training recipes the frame based cross-entropy objective has been replaced by the so-called chain objective. Scaling the targets is unfortunately not feasible for a chain objective (or would be so complex to make it that it cannot be tolerated even during training), but we still have the opportunity to use KLD regularization as part of the regularization in place already: the chain objective D_{chain} may be itself regularized by interpolating it with the cross entropy based framewise objective D_{fCE} (this is even supported by Kaldi through the *output-xent* output layer):

$$D_{chain+fCE} = (1-\beta)D_{chain} + \beta D_{fCE} \tag{3}$$

By extending the objective to

$$D_{chain+fCE+KL} = (1-\beta)D_{chain} + \beta D_{KL} \tag{4}$$

we can observe a frame based regularizing term. Technically such regularization is carried out by preserving a copy of the original model (the source model) and use it in parallel to the newly trained branch of network.

4 Analysed Adaptation Approaches

4.1 Vanilla Retraining

Retraining is probably the simplest way of adapting a model. For retraining, we use target domain data in the same manner we use the source domain data for training the source model, with the exception that retraining has to be more conservative. The topology of the source model is not adapted, which means that the senone representation reflects knowledge extracted essentially from the source domain data. This has advantages and disadvantages: the pro side is that we can preserve a robust tree estimate which would be hard to achieve on the low amount of target domain data; the negative side however, is that the tree may be mismatched if context dependency in the target domain differs too much from the source domain. Unfortunately, this is an often case, to reduce the negative effects, the number of senones can be reduced so that we use a broader model that has higher chance to match the target dialect better. What is the optimal number of senones, however, has to be determined experimentally on a development set.

To prevent overfitting, the primary interest is to use low learning rates, which we will set layerwise so that we let the top layers learning more. The secondary way to regularize retraining consists in increasing the weights α and β in the regularization terms of the objective $D_{chain+fCE+KL}$, as explained in subsection 3.3. In the supervised adaptation scheme, alignments for the adaptation data can be generated with the source model, but the source model can also be used to decode the target data in the first step and use the so-obtained lattices as soft alignments for the retraining phase. Using lattices fits perfectly into the chain training of TDNN acoustic models with the Lattice Free Maximum Mutual Information (LF-MMI) objective, as numerator lattices are directly available from the decoder.

4.2 Transfer Learning

The transfer learning approach is also based on retraining, but the model structure is not preserved in this case [15]. As the SoftMax output layer represents the phone context dependency tree, this means that both have to be trained from scratch, based on target domain data. At least the last hidden layer of the source model has also to be removed, as the number of output nodes will be likely different compared to the source model. These layers are newly initialized, and their parameters are learned from scratch by training the modified model on target domain data. The context dependency tree is obtained as resulting from training a GMM based acoustic model on the target domain data. The same GMM can then be used to generate alignments for supervised adaptation. Compared to simple retraining, the advantage is that the output is tuned according to the target domain statistics. The drawback is that this tuning has to be carried out on relatively low

amount of data in case of adaptation, which is a risk factor of getting a somewhat mismatched senone layout. Using unsupervised adaptation gets mostly out of the scope as well, as for training the small target domain GMM model transcriptions are desirable, but even if transcription is automated with the source domain decoder, using a lattice is not as straightforward as with the unchanged topology in simple retraining.

4.3 Factorization

As outlined earlier, factorization-based adaptation approaches offer an alternative for adaptation scenarios with very low amount of target domain data available [17, 24]. In this work we use the Singular Value Decomposition (SVD) algorithm and decompose the weight matrix W between two particular layers as follows:

$$W_{m \times n} = U_{m \times k} S_{k \times k} V_{k \times n}. \tag{5}$$

Factorization goes hand-by-hand with modifying the network structure, but this is a non-invasive action compared to the reinitialization of the complete top layers required for the transfer learning approach. With SVD, the output senone layout is not changed. Usually a single bottleneck is created (or an existing bottleneck is used) in the neural acoustic model. The dimensions of the bottleneck allow for a fine control over the number of parameters to be retrained. Comparing SVD to simple retraining, we select or create a small subset of latent network parameters and freeze all other (meaning a zero-learning rate).

The advantages of the factorization approach include the scalability and aptitude to work with low amount of data, whereas its drawbacks can be summarized as limiting the adaptation to a small part of the parameters which allows for only slight shifts in model outputs. Both supervised and unsupervised adaptation schemes can be suitable as the output layer is not affected by the adaptation.

5 Research Hypotheses

One of our goals in this work is to compare different acoustic model adaptation approaches for ASR. As we discussed in the introduction, several approaches in several flavours have been proposed for acoustic model adaptation, but most of them were evaluated within specific setups and special conditions, making a fair comparison intractable. Therefore, we will use a common baseline setup and the same environment for all experiments, with the main research questions condensed around which approach could be suitable given an hour or some ten hours of adaptation (target) data. Our research questions and null hypotheses are as follows:

- Can factorization-based approaches outperform vanilla retraining in the case if very low amount of target data is available for adaptation? – We hypothesize that vanilla retraining is better as factorization introduces loss through the used bottleneck (H1).
- Can transfer learning outperform vanilla retraining in the case if higher amount of adaptation data is available? Our hypothesis is that after reaching a threshold transfer learning yields better results since it has a better matching output layout (H2).
- Can we leverage KLD regularization for better convergence in the case if we combine it with cross entropy-based regularization for chain models? We hypothesize that KLD regularization improves the convergence (H3).

6 Experiments

6.1 Configurations

We perform the source model adaptation with the 3 different strategies by gradually adding more target data and by also gradually performing more and more training epochs with dynamically decreasing learning rate. We test the ASR performance in terms of Word Error Rate (WER) on the held-out test set.

We use and adopt the Kaldi Librispeech recipe for all kinds of adaptation experiment. Alignments on the target data are generated with the Librispeech `tri6b-cleaned' model (the same used for training the source acoustic model TDNN) for supervised adaptation. In case of unsupervised adaptation, we use the default offline Librispeech Kaldi decoder with its default settings (for the 1c recipe as of October 2018) and the 'tgsmall' language model. We did not perform second pass rescoring for the obtained lattices. For the individual approaches we use the following configurations:

- for transfer learning, we train the GMM acoustic model and its associated phone context decision tree on the target data and we reduce the target number of tree leaves from 7k to 1.8k. (No further fine tuning for the number of leaves was investigated depending on the amount of the target data.)
- for factorization, we create the bottleneck after the last TDNN layer and before the 'prefinal' layer with 256 dimensions, and initialize it following [17] (S is initialized as an identity matrix). In the recent Kaldi recipes with factorized TDNN architecture (F-TDNN), we can simply add a linear layer initialized as an identity matrix for this purpose.
- for vanilla retraining we tune α and β in the range of [0,1], letting them to sum to 1 in each case.

6.2 Number of Iterations

Plotting WER from epoch to epoch for a vanilla retraining illustrates in Fig. 1 as the model is being shifted toward the target domain: we observe reducing WER on target evaluation data and increasing WER on source domain test data. Most of the performance gain is obtained within the first three iterations (1 iteration contains 1.5 M chain examples in the used setup).

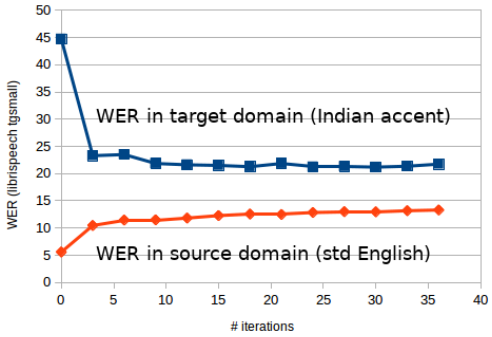

Figure 1
Illustration of the adaptation process with simple retraining (WER in %) using 10 hours of Indian English adaptation data

Final reduction on WER in the target domain is as high as 50% relative, although the basis is also high, given the Indian accent is far from the US English one. It is not worth running more than 18 iterations with this setup using 10 hours of Indian accent data and 1.5 M frames per iteration (note, this depends both on minibatch size and the amount of available adaptation data), as there is no more performance gain, but models may overfit, i.e., the WER seen with source data keeps worsening. In parallel, robustness of the AM is likely to become worse.

6.3 Amount of Data

Taking again the example of Indian English adaptation of the US English source model, Figs. 2a-c show convergence seen on eval data for all the 3 adaptation scenarios and gradually adding data of 1k, 2k, 5k and 10k utterances, corresponding roughly to 1 hour, 2 hours, 5 hours and 10 hours of target domain data, respectively. We can observe, that – according to reasonable expectations – the more data we use, the higher WER reduction we obtain; and that the more data we have, also the more iterations we have to run during the retraining phase.

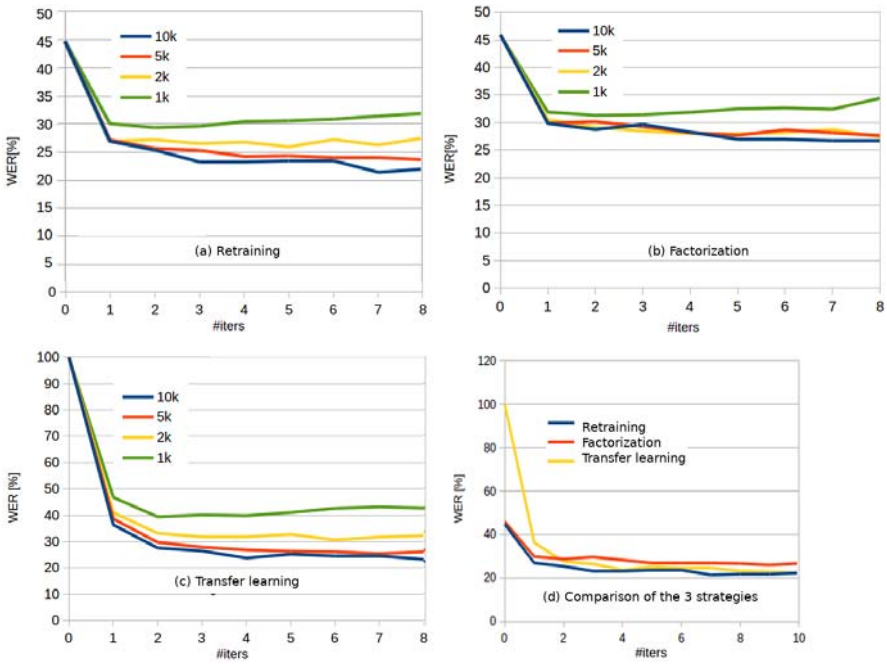

Figure 2

Illustration of the adaptation process with the 3 strategies: (a) simple retraining (upper left), (b) factorization (upper right) and (c) transfer learning (bottom). Pane (d) shows the comparison of the 3 methods (on 10 hours of Indian accented English).

Fig. 2b shows the factorization adaptation outcome, where we see a performance gain saturating at 2 hours of adaptation data and adding further data does not result in additional WER improvement. This can be explained partly by the fixed bottleneck size at 256 nodes irrespective of data quantity, but as both other strategies overperformed the factorization approach already by smaller amount of data, we did not find it worthwhile to increase bottleneck dimensionality. This also means that albeit the available amount of data is usually considered a key parameter when choosing the adaptation strategy, i.e., factorization alike approaches are reported to have the advantage of scalability [17] and work well with low amount of target data, we still found vanilla retraining outperforming it by using equal amount of data. This corresponds to confirming our H1 (by 5% confidence). Even with datasets containing only 15 and 30 minutes of speech and despite scaling the bottleneck size down to 128, we were not able to demonstrate such difference between the approaches. Using i-vectors in the baseline system may interplay in this result, but as i-vectors are regarded to be state-of-the-art, we found it pointless from a practical perspective to experiment with discarding them.

6.4 Effect of Top Layout Adaptation

By using all the 10 hours of Indian development data we compared the convergence of the 3 strategies in Fig. 2d. Still the vanilla retraining approach is the best performing one. We can therefore conclude that for the used 10 hours of data, the advantages of a better fitting output layer cannot counteract the benefit of having a somewhat mismatched output layer, which was trained on more data even it is out-of-domain data. Looking at Table 1 we observe however, that for other dialects associated with more training data, adapting the top layout can yield a better model than the one obtained with vanilla retraining, as is shown in the case of English spoken in England. On British English with 31 hours of adaptation data, transfer learning performs better. The case of US English may seem contradictory to this finding, as with even higher amount of 59 hours of adaptation data we again observe vanilla retraining yielding a better model, but we resolve this contradiction with the following arguments: the source model is trained on the Librispeech dataset, which contains dominantly US English utterances. The domain mismatch therefore does not arise from different variants of the same language, but rather from a mismatch in the acoustic conditions of the speech data used for adaptation, taken from the US English samples of the Mozilla Common Voice database. We argue that as dialect mismatch should be minimal between the two corpora, the worse performance of the adjusted topology model can rather be attributed to the poorer modelling capabilities of the context dependency decision tree, which, trained on less data can provide a poor representation for the particular dialect.

Table 1

WER reduction obtained with different supervised adaptation strategies

	AUS	CDN	ENG	IND	USA
Amount of adaptation data (hours)	8	8	31	10	59
Baseline WER	25.3	13.3	19.9	44.8	24.5
WER reduction vanilla retraining	**35.2**	**20.3**	31.7	**52.1**	**46.9**
WER reduction top layer reinitialization	28.9	12.0	**35.2**	50.3	44.1
WER reduction factorization	-	-	-	42.0	-

The difference in the relative WER improvement with the transfer learning and the retraining strategies is significant in the case of the ENG variety (by 5% confidence level), so our H2 hypothesis is confirmed. However, the threshold point for the amount of data when reinitialization of the top layer becomes beneficial does not only depend on the number of hours of the target data: it may hence be worth considering on a case-by-case basis that transfer learning requires an additional precursory step to architect the output layout by training a GMM on the adaptation data.

6.5 Learning Rate and Regularization

We applied a factor of *0.05..0.5* to the default learning rates worked out by the Kaldi TDNN trainer. On 10 hours of adaptation data we consistently experienced that the learning rate factors between 0.1 and 0.15 work the best. Letting the learning rate be higher (up to 0.5) we observed a consistent increase in WER after adaptation leading to by 5-6% less relative improvement in WER.

We were experimenting with applying KL regularization as part of the cross-entropy regularization. Although the difference between using simply the $D_{chain+fCE}$ objective or the $D_{chain+fCE+KL}$ objective is not large, we experienced smoother convergence and slightly better performance with $D_{chain+fCE+KL}$, the difference remaining however below significance (by 5% confidence, we obtained the best results with $\alpha=0.3$ and $\beta=0.6$). We hence had to reject our H3. Using KL-regularization is reportedly beneficial when training with cross-entropy objective [30, 31]. Using this approach for regularizing chain training was hence not significantly better; we had the impression that in part regularizing with cross-entropy (without KL) is already effective enough, and in part that the backstitch training [28] has a similar regularizing effect than has adding KL-regularization. We did not carry out targeted experiments to do further research on these, however.

7 Unsupervised Adaptation

In several cases there may be no available gold transcription for the utterances which nevertheless could be valuable for acoustic (or language) model adaptation. By providing an unsupervised adaptation scheme, systems can make a step toward self-learning. Obviously, for training, a transcription has to be provided in any case. In the unsupervised case, this transcription is obtained by decoding with the existing baseline model. The source model may also be a specific model which needs further adaptation (incremental adaptation).

For unsupervised adaptation, the quality of transcriptions is never good enough, otherwise the adaptation would not be required. Therefore, a mechanism allowing for either predicting the accuracy (or confidence) of the transcripts, or soft target training to represent multiple hypotheses can be very useful to carry out the adaptation on reliable data.

7.1 Predicting Confidence

The ASR itself is able to produce confidence in a normalized range associated with words in the decoded lattice. These confidences result from the acoustic and the language model scores, hence are less favourable for acoustic model

adaptation, because if the grammar used for decoding is poorly fitting, confidences get heavily biased. Analysing some recordings along with their confidence lattices has revealed that the predicting power of the confidence scores for telling whether a word is correctly recognized is too low for a meaningful exploitation in our task. Indeed, providing confidence is known to be a hard task, no standalone feature is known with strong predicting power, therefore, an ensemble of weak classifiers was considered an alternative: TranscRater [32] is a freely available toolkit building upon an abundant feature set to predict WER, and the prediction is independent from ASR confidence and does not require reference (supervision) for inference. The input features rely on four main categories: speech signal (MFCC plausibility), lexicon (how difficult is to transcribe a given word), language model (several can be used, even different from the one used for decoding, word likelihoods and perplexities are extracted) and part-of-speech (to estimate whether the observed syntax makes sense). As said, there is no confidence score used from the ASR, but TranscRater has to be trained prior to using it, which however requires a supervised subset, or at least a correct (high precision) value for the WER. This is obviously a considerable drawback.

While evaluating TranscRater capabilities for our task, we found a very weak correlation between the actual and predicted WER on our development sets, therefore, we dropped using it. We were also experimenting with predicting the M-measure [33], which evaluates acoustic consistence and plausibility and therefore it can be a promising candidate to prepare acoustic model adaptation, as it considers only the acoustic aspects of data. In a similar framework to M-measure, it is also possible to use acoustic model posteriors for WER prediction (or provide a score for the goodness of the transcript), which completely eliminates the dependence on the HCLG (and hence language model). Without representing here, the exact formulae, the idea behind M-measure is to capture posterior divergence of the acoustic model for several time spans and compute the mean of them. Divergence is computed as the symmetric Kullback-Leibler distance between posterior distributions yielded by the acoustic model for a certain time. Intuitively, this term penalizes smooth distributions (where there is no outstanding posterior probability associated with a certain senone), and also penalizes inconsistency over time, i.e. a recurrent senone should be consistently represented at the posterior level, which could minimize the KLD and hence M-measure, too. M measure was found to yield better estimates than ASR confidence or TranscRater, but still below a fair correlation between the M-measure and the WER.

7.2 Soft Targets with Lattice Supervision

An alternative to predicting confidence (reliability of transcripts) is to provide a soft transcript, which allows for multiple hypotheses. A lattice is an ideal choice instead of a hard transcript. Using a lattice moreover matches perfectly with

discriminative training strategies and given the computation load reduction resulting from applying a lattice free denominator, the numerator can be allowed to remain more complex.

We prepared experiments by using standard decoder settings (in terms of beam and other respective parameters) with the Librispeech TDNN baseline and the 'tgmed' language model. The first task is augmenting the data: a speed and volume perturbation-based augmentation can be used, followed by extraction of high resolution (hires) features. Thereafter, the decoder can be run by preserving the produced lattices for the numerator in TDNN discriminative training.

Adaptation results are summarized in Table 2. Results suggest that the closer the accent is to the standard one, the less amount of data is required, and that the more target data we have, the more powerful becomes creating a new layout (last hidden layer, SoftMax and output based on target domain state tying tree).

Table 2
WER reduction obtained with unsupervised adaptation

	AUS	CDN	ENG	IND	USA
Amount of adaptation data (hours)	8	8	31	10	59
Baseline WER	25.3	13.3	19.9	44.8	24.5
WER reduction vanilla retraining	**16.4**	11.3	**15.4**	33.3	**16.2**
WER reduction top layer reinitialization	**16.4**	**11.7**	13.3	**34.1**	15.4

Compared to the supervised case, the obtained WER reduction is lower, but still over 10% in all cases. Changing the output layout was slightly better performing than vanilla retraining, probably because training a new context dependency tree was better fitting to automatic transcripts (lattices) containing errors. As a rule of thumb, we may suggest using a new context dependency tree and adjusted model topology for unsupervised adaptation, or stay with vanilla retraining when adaptation data is in the range of 10 or a few tens of hours.

Conclusions

In this paper we have addressed dialect adaptation of the ASR acoustic models. We compared vanilla retraining without network topology changes to two alternatives involving topology changes: factorization between hidden layers and creating new senone layout by preserving lower network structure with transfer learning. We found that even with low amount of adaptation data available, factorization alike approaches underperformed vanilla retraining. Regarding transfer learning with changing the top layout, we obtained better results than with vanilla retraining if adaptation data was abundant. Adding cross-entropy based Kullback Leibler regularization did not improve significantly the WER after adaptation, as other regularizing component and techniques – using cross-entropy and backstitch – are likely to have a similar effect. Our overall conclusion can be summarized as follows: instead going for sophisticated approaches, the more

traditional solutions work also well by much lower complexity and requiring less effort.

References

[1] P. Baranyi and A. Csapo, Cognitive infocommunications: coginfocom, in 2010 11th International Symposium on Computational Intelligence and Informatics (CINTI) IEEE, 2010, pp. 141-146

[2] A. Csapo and P. Baranyi, An adaptive tuning model for cognitive info-communication channels, in 2011 IEEE 9th Int. Symposium on Applied Machine Intelligence and Informatics (SAMI) IEEE, 2011, pp. 231-236

[3] L. Czap and L. Zhao, Phonetic aspects of Chinese Hhaanxi Xi'an dialect, in 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2017, pp. 000 051-000 056

[4] G. Szaszák and P. N. Garner, Evaluating intra-and crosslingual adaptation for non-native speech recognition in a bilingual environment, in 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2013, pp. 357-362

[5] J. Galić, B. Popović, and D. Š. Pavlović, Whispered speech recognition using hidden markov models and support vector machines, Acta Politechnica Hungarica, Vol. 15, No. 5, 2018

[6] A. López-Zorrilla, M. de Velasco Vázquez, S. Cenceschi, and M. I. Torres, Corrective Focus Detection in Italian Speech Using Neural Networks, Acta Polytechnica Hungarica, Vol. 15, No. 5, 2018

[7] Y. Kiryu, A. Ito, and M. Kanazawa, Recognition Technique of Confidential Words Using Neural Networksin Cognitive Infocommunications, Acta Polytechnica Hungarica, Vol. 16, No. 2, pp. 129-143, 2019

[8] I. Poggi, F. DErrico, and L. Vincze, Uncertain Words, Uncertain Texts. Perception and Effects of Uncertainty in Biomedical Communication, Acta Polytechnica Hungarica, Vol. 16, No. 2, 2019

[9] M. Rusko and M. Finke, Using speech analysis in voice communication: A new approach to improve air traffic management security, in 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom) IEEE, 2016, pp. 000 181-000 186

[10] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, arXiv preprint arXiv:1312.6211, 2013

[11] D. M. Hawkins, The problem of overfitting, Journal of chemical information and computer sciences, Vol. 44, No. 1, pp. 1-12, 2004

[12] 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, in 4th Conference on Speech Communication and Technology, 1995

[13] 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. 10-11, pp. 827-835, 2007

[14] P. Swietojanski, J. Li, and S. Renals, Learning hidden unit contributions for unsupervised acoustic model adaptation, IEEE/ACM Trans. on Audio, Speech and Lang. Proc., Vol. 24, No. 8, pp. 1450-1463, 2016

[15] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, Investigation of transfer learning for ASR using LF-MMI trained neural networks, in Automatic Speech Recognition and Understanding Workshop (ASRU) 2017 IEEE. IEEE, 2017, pp. 279-286

[16] S. Parveen and P. Green, Multitask learning in connectionist robust asr using recurrent neural networks, in Eighth European Conference on Speech Communication and Technology, 2003

[17] Y. Zhao, J. Li, and Y. Gong, Low-rank plus diagonal adaptation for deep neural networks, in Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016, pp. 5005-5009

[18] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, Speaker adaptation of neural network acoustic models using i-vectors, in ASRU, 2013, pp. 55-59

[19] R. Doddipatla, N. Braunschweiler, and R. Maia, Speaker Adaptation in DNN-Based Speech Synthesis Using d-Vectors, in INTERSPEECH, 2017, pp. 3404-3408

[20] Tripathi, A. Mohan, S. Anand, and M. Singh, Adversarial learning of raw speech features for domain invariant speech recognition, in Acoustics, Speech and Signal Processing (ICASSP) 2018 IEEE International Conference on. IEEE, 2018

[21] V. Manohar, P. Ghahremani, D. Povey, and S. Khudanpur, A Teacher-Student Learning Approach for Unsupervised Domain Adaptation of Sequence-Trained ASR Models, in 2018 IEEE Spoken Language Technology Workshop (SLT) IEEE, 2018, pp. 250-257

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., The Kaldi speech recognition toolkit, in IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011

[23] V. Peddinti, D. Povey, and S. Khudanpur, A time delay neural network architecture for efficient modeling of long temporal contexts, in Sixteenth Annual Conference of the International Speech Communication Association, 2015

[24] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks, in Interspeech 2018, 2018, pp. 3743-3747

[25] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, X-vectors: Robust DNN embeddings for speaker recognition, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2018, pp. 5329-5333

[26] D. Yu and L. Deng, Adaptation of deep neural networks, in Automatic Speech Recognition. Springer, 2015, pp. 193-215

[27] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, Backstitch: Counteracting finite-sample bias via negative steps, In: Interspeech 2017, pp. 1631-1635

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in Acoustics, Speech and Signal Processing (ICASSP) 2015 IEEE International Conference on. IEEE, 2015, pp. 5206-5210

[29] The Mozilla Common Voice corpus, https://voice.mozilla.org/, 2017

[30] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition, in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013, pp. 7893-7897

[31] D. Falavigna, M. Matassoni, S. Jalalvand, M. Negri, and M. Turchi, DNN adaptation by automatic quality estimation of ASR hypotheses, Computer Speech & Language, Vol. 46, pp. 585-604, 2017

[32] S. Jalalvand, M. Negri, M. Turchi, J. G. de Souza, D. Falavigna and M. R. Qwaider, Transcrater: a tool for automatic speech recognition quality

[33] B. T. Meyer, S. H. Mallidi, H. Kayser and H. Hermansky, *Predicting error rates for unknown data in automatic speech recognition*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2017