Multitask Learning of Context-Dependent Targets in Deep Neural Network Acoustic Models

Multitask learning of context-dependent targets in deep neural network acoustic models

Peter Bell, *Member, IEEE,* Pawel Swietojanski, *Member, IEEE,* and Steve Renals, *Fellow, IEEE*

***Abstract*—This paper investigates the use of multitask learning to improve context-dependent deep neural network (DNN) acoustic models. The use of hybrid DNN systems with clustered triphone targets is now standard in automatic speech recognition. However, we suggest that using a single set of DNN targets in this manner may not be the most effective choice, since the targets are the result of a somewhat arbitrary clustering process that may not be optimal for discrimination. We propose to remedy this problem through the addition of secondary tasks predicting alternative content-dependent or context-independent targets. We present a comprehensive set of experiments on a lecture recognition task showing that DNNs trained through multitask learning in this manner give consistently improved performance compared to standard hybrid DNNs. The technique is evaluated across a range of data and output sizes. Improvements are seen when training uses the cross entropy criterion and also when sequence training is applied.**

***Index Terms*—automatic speech recognition, multitask learning, context modelling**

I. Introduction

THE ability to effectively model phonemes in context is critical to the good performance of modern automatic speech recognition (ASR) systems for continuous speech. The hidden Markov model (HMM) based approach that remains the dominant paradigm for ASR explicitly makes the assumption that the acoustic observation for each frame is dependent only on the current hidden state, implying that adjacent observations are conditionally independent, given the hidden state sequence. This assumption is of fundamental importance for efficient decoder design.

This model has two obvious weaknesses. First, the frame-wise conditional independence assumption is clearly incorrect. We term this the *acoustic context* problem; it can be partly addressed by appending first and higher order temporal derivative features to the acoustic feature vectors, by various forms of recurrent structures, or by the use of wide-context windows. This assumption also necessitates the use of an scaling factor for the acoustic probabilities [1].

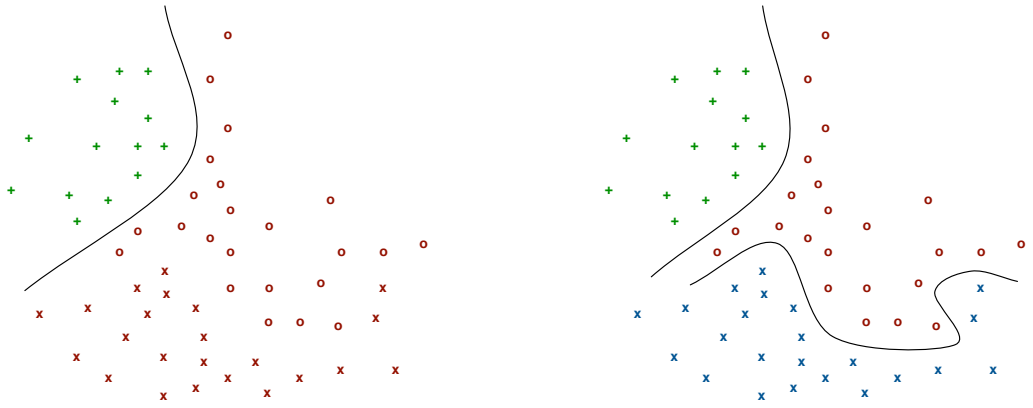
The second weakness, which can be termed the *phonetic context* problem is that, due to the physical constraints of the human articulatory system, the realisation of a phoneme is highly influenced by the adjacent phonemes. This effect, known as co-articulation, is particularly prevalent in faster, more natural speech. Within the HMM framework, this effect is primarily modelled by the use of context-dependent (CD) phone models [2], [3], [4] as the basis for acoustic modelling, contrasting with the original HMM systems which used context-independent (CI) monophone models [5], [6]. In essence, this allows the decoder to dynamically adapt the probability distribution for each monophone according to its context within the hypothesised phone sequence. Unfortunately, modelling each phone with both left and right context – known as a triphone – results in a very large number of states to model (for example, up to $3 \times 48^3 \approx 300,000$ in a typical English system using 3-state HMMs). Furthermore, due to the uneven distribution phones in speech, many of these triphones will be unseen in training data. An important innovation in HMM-based ASR was the introduction of phonetic decision trees to cluster the CD state units [4]. These tied states (also known as *senones* [7]) were modelled using Gaussian mixture models (GMMs). By tying triphones in this way, data sparsity issues are reduced, and at decoding time the decision tree can be used to select a probability distribution for unseen triphones. Tied-state modelling in HMM-GMM systems was a contributing factor in their outperforming neural network based approaches [8], [9] used at the time, and these CD-HMM-GMM systems formed the foundation of state-of-the-art ASR systems for the following fifteen years.

The use of deep neural network (DNN) acoustic models [10], [11] has led to significant improvements in the accuracy of speech recognition systems. DNN acoustic models are typically trained to predict posterior probabilities over CD tied states derived from clustering with a previously-trained GMM. Outputs from these DNNs can be used directly in a standard CD-HMM decoder, after scaling by a prior to convert the posterior probabilities to pseudo-likelihoods [12]. This "hybrid DNN" approach is now standard in modern systems, and while GMM systems are still routinely used to obtain the CD state clustering, other purely DNN-based methods may also be used [13], [14]. Neural network models which take alternative approaches to solve the acoustic context problem, such as recurrent neural networks (RNNs) [15] and LSTMs [16], [17], still generally use the same tied-state solution to the phonetic context problem, although recent work [18] has used the CTC loss function defined over CI states.

Context-dependent state-tying contrasts with earlier efforts to incorporate context modelling in neural network systems, where context information was incorporated by means of a bias term in the hidden layer [19]. A more recent alternative [20] factorises the left and right acoustic contexts using multiple sets of articulatory-based equivalence classes,

P Bell, P Swietojanski, and S Renals are with The Centre for Speech Technology Research, University of Edinburgh, UK; email: {peter.bell,p.swietojanski,s.renals}@ed.ac.uk

Supported by EPSRC Programme Grant grant EP/I031022/1 *Natural Speech Technology* (NST) and the European Union under H2020 project *SUMMA*, grant agreement 688139. The NST research data collection may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786.



(a) Same class (b) Different classes

Fig. 1. A theoretical visualisation of the problems in modelling tied-state units with a DNN: in (a), two triphones 'o' and 'x' of the same monophone are easily discriminated from triphone '+' of another monophone. In (b), however, the two triphones happen to be in different tied-state classes, and DNN is forced to learn to discriminate between the two.

training DNNs for each scheme, and allowing probabilities for each unique triphone to be synthesised by combining the appropriate DNN probabilities using a regression model.

We suggest that the standard state-tying approach in DNN acoustic model has significant deficiencies. When used to generate targets for generative models such as GMMs, modelling CD units can be viewed as a more fine-grained partitioning of the acoustic space to allow larger, more powerful models compared to CI units, whilst the use of state-tying simply balances the requirement for more precise modelling of context with the need to ensure that there is adequate data for reliable estimation of the GMM parameters for each state. However, the situation is quite different for an inherently discriminative model such as the DNN, where only the labels of boundary cases determine the decision boundaries. This leads to a problem: because the targets derived from the clustering scheme are in a sense arbitrary, depending on the heuristically-chosen number of leaves in the decision tree and the question set used, two samples which the DNN is trained to discriminate between under one clustering scheme could be assigned the same label in a different scheme. The DNN is effectively over-fitting to a poorly-defined labelling scheme.

A second problem, as noted in [19], is that the primary requirement of the DNN is to discriminate between monophones, with the left and right targets used as an adjustment to better model the monophones in context. However, in the state-tying scheme the costs of misclassification between senones of the same monophone and senones of different monophones are treated equally. This could conceivably result in the DNN "wasting" parameters – attempting difficult discriminations that may not be useful at test time – and could result in lower layers learning poor-quality representations of the data. A theoretical visualisation of this situation is illustrated in Figure 1.

Motivated by these concerns, we propose to use multitask learning (MTL) as a technique to regularise the DNN, preventing it from over-fitting to a single set of senone targets and thus learning better hidden representations of the data. This is achieved by forcing the network to learn additional CI or CD labels as well as the conventional senone targets. This paper extends our previous work [21], [22] and performs further analysis of the findings. As a means of solving both the above problems, we first consider using the prediction of monophone states as a secondary task for a standard CD-DNN. We investigate the effects of this technique when both the quantity of training data and the number of senones is varied. We also investigate the improvements in accuracy from applying sequence training.

Second, we investigate the use of alternative CD secondary tasks, attempting to verify the hypothesis that simply over-fitting to a single set of targets is harmful. We also seek to demonstrate that the improvements from the use of monophones as a secondary task are not simply due to the lower cardinality of the task. Finally, we perform experiments using alternate implementations of MTL.

II. Multitask learning

Multitask learning [23] expresses the general principle that machine learning models designed to solve different problems on the same data can beneficially share some common representation.

In DNNs MTL may be implemented by creating a network with shared hidden layers. In this context a task, A, is effectively a mapping from a set of T training frames to a set of labels, that is:

$$\begin{aligned} A: \ \{t: 1 \leq t \leq T\} &\mapsto \{1, \ldots, |A|\} \\ t &\mapsto y_t^A \end{aligned} \tag{1}$$

where $\mathbf{x}_t$ is the data, y_t^A its labelling under task A, and $|A|$ denotes the cardinality of the task. Then we can define the objective for task A by the cross-entropy

$$\mathcal{F}_A(\theta) = \sum_t^T \log p(y_t^A | \mathbf{x}_t; \theta) \tag{2}$$

which is maximised with respect to parameters θ when learning task A.

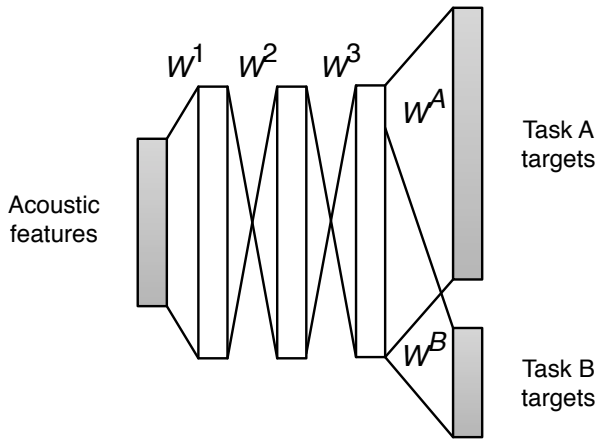


Fig. 2. An example multitask DNN architecture

When applying multi-task learning, we introduce an additional task B, which defines an alternative labelling on the same set of T training frames. Simultaneous to optimising $\mathcal{F}_A(\theta)$, we now additionally attempt to optimise $\mathcal{F}_B(\theta)$. It is possible to extend this to multiple additional tasks, although we typically consider a single additional task in this work.

In the multi-task experiments that we reported in this paper, we use only the primary task at test time, denoted as task A. The use of a single primary task contrasts with related approaches in which multiple tasks are combined at test time [24], or with situations where separate tasks are useful in their own right – for example, multi-lingual systems, where different tasks correspond to the phone sets of different languages [25].

A. Multitask learning in deep neural networks

Consider a feed-forward network, with L hidden layers $\{\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)}\}$:

$$\mathbf{h}_t^{(\ell)} = \sigma(\mathbf{W}^{(\ell)}\mathbf{h}_t^{(\ell-1)} + \mathbf{b}^{(\ell)}) \tag{3}$$

where σ is the sigmoid non-linearity. Given the activations in the final hidden layer, the network produces separate outputs for each task. For task A, these are given by

$$\mathbf{z}_t^A = \mathrm{softmax}(\mathbf{W}^A\mathbf{h}_t^L + \mathbf{b}^A) \tag{4}$$

with weight W^A and bias $\mathbf{b}^A$ specific to this task (we implicitly drop the layer-related index ℓ). $\mathbf{z}_t^A$ is the vector of posterior probabilities for the labels of task A. An example network with two tasks is shown in Figure 2. All networks used in this paper share all hidden layers, with only the output layers being task-specific. However, it is easy to make some of the hidden layers task dependent also.

When performing an update for task A, we obtain the gradient of the cross-entropy criterion for task A, with respect to all parameters θ:

$$\frac{\partial \mathcal{F}_A(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\sum_t \log p(y_t^A|\mathbf{x}_t;\theta) = \sum_t \frac{\partial \mathbf{z}_t^A(y_t^A)}{\partial \theta} \tag{5}$$

The derivatives are computed using the back-propagation algorithm. The only zero-gradients are for the output-layer parameters of other tasks, ie. $\mathbf{W}^B$, $\mathbf{b}^B$. The sums are computed over mini-batches.

An important consideration is how the primary and secondary task updates are interleaved. This could be done at the minibatch level. If a weight λ is assigned to task A, then the update for the minibatch is given by

$$\frac{\partial}{\partial \theta}\left[\lambda\mathcal{F}_A(\theta) + (1-\lambda)\mathcal{F}_B(\theta)\right] = \sum_t \left[\lambda\frac{\partial \mathbf{z}_t^A(y_t^A)}{\partial \theta} + (1-\lambda)\frac{\partial \mathbf{z}_t^B(y_t^B)}{\partial \theta}\right] \tag{6}$$

This method is computationally efficient, since when computing the gradients, the forward pass can be shared between both tasks, up to output layer. However, we postulate that this approach lacks efficiency from a learning perspective since one task is typically strongly connected to the other, leading to a risk that the model learns the correlation between outputs of each task, rather than learning a representation that is effective for both tasks independently. This motivates an alternative implementation used in this work: we interleave updates to optimise $\mathcal{F}_A(\theta)$ and $\mathcal{F}_B(\theta)$ at the minibatch level, but present minibatches for the tasks randomly, so that the two updates are not related in any way. We experimentally compare the two methods in Section IV-F.

B. Explaining multitask learning

There are a number of possible explanations for the gains obtained from multitask learning [23], which we investigate experimentally in this paper. A shared representation between tasks is central to the multitask approach. To learn a model with good generalisation capabilities, we seek a low-dimensional hidden representation which is sufficient to explain the labels of interest, whilst reducing noise in the data [26]. The theory is that we should have a preference for representations that are capable of explaining multiple properties of data. Therefore, by providing additional label information for each sample, a second task can enable a better hidden representation to be learned.

However, in our proposed use of MTL, the secondary task – whether it is simply the monophone label or a lower-cardinality senone set – does not provide additional information over the primary task labelling. If MTL is effective here, it would suggest that alternative explanations should be considered, for instance that the use of an additional task reduces the variance in the error signals, simply by averaging over more sample/label pairs. If a secondary task has a lower cardinality – or lower entropy – than the primary task, then it is similarly likely to be easier to learn, allowing reliable convergence to a good hidden representation. This is similar to the motivation for curriculum learning [27], where the entropy of the task is gradually increased as learning progresses. It leads to the notion of *eavesdropping*, whereby a hidden representation may be hard to learn from task A but easier from task B, so that in a multitask setting, A is able to "eavesdrop" [28], [23] on the information provided by B.

Specific to the modelling of triphones, as discussed in Section I, it is possible that in attempting to bias the model towards predicting a context-independent secondary task, it is also learning a discrimination that is more useful to the test time task. In this case, we might expect these gains to reduce after the application of sequence training [29], [30], which explicitly achieves this aim by optimising a criterion related to the expected error over the complete utterance. Alternatively, it could be that gains arise by reducing over-fitting to a single set of senone targets which, although yielding a more fine-grained division of the acoustic space, are essentially an arbitrary product of the clustering process. We suggest that the use of MTL result in a "goldilocks" scenario – a "just right" balance of a monophone task that is well-defined but not informative enough to guide to the model to a good hidden representation, and a high-dimensional task that provides detail, but where the labels, and hence the training signals, have a high degree of statistical noise.

C. Related work

Multitask learning was first applied to speech recognition in situations in which the primary task was monophone classification. In the context of robust speech recognition using a hybrid RNN/HMM architecture, Parveen and Green [31] used an MTL approach in which the secondary task was speech enhancement. Stadermann et al [32] investigated a number of secondary tasks: gender classification, broad phonetic class recognition, and grapheme recognition, for clean speech recognition again using a hybrid RNN/HMM system.

More recently Seltzer and Droppo [33] investigated an approach in which DNNs trained for monophone recognition on TIMIT were augmented with a variety of context-related tasks. In contrast to the current work, however, no context was modelled in the primary task, making the baseline model relatively weak, with adding significant additional information being added via the additional tasks.

Chen at al [34] employed MTL as a regulariser in ASR by modelling triphones jointly with trigraphemes. Zhang and Woodland [35] investigated the use of the lower-dimensional monophone states as targets for disciminative pre-training, as an initialisation for context-dependent DNNs. This can be viewed as a form of curriculum learning.

We proposed a monophone classification secondary task for CD phone modelling [21], showing that this approach outperforms the monophone pre-training approach of Zhang and Woodland. We also found that gains from using MTL in this way continue to hold when used in combination with curriculum learning – when monophone pre-training is used to initialise the networks, prior to fine-tuning with MTL. However, we did not find consistent gains from curriculum learning when it was used in combination with MTL in this manner.

Chen et al [36] reported similar gains from the use of MTL with a monophone secondary task in a scenario with a very large number of distinct triphone models. A similar approach has been subsequently investigated by Siohan and Ryback [24] who found that the use of alternative decision trees yielded benefits in system combination settings, but did not provide benefits from learning multiple sets of targets simultaneously in a multitask setting, in contrast to the findings reported here. This may be due to the relatively small number of tied states (12,000) relative to the quantity of data (2,000hrs) used in [24]. This has been recently followed by the use of "meta-states" comprising tuples of states from different decision trees [37].

III. Experimental setup

We carried out experiments on the English TED Talk transcription task from the IWSLT evaluation [38], presenting results on the `dev2010`, `tst2010` and `tst2011` test sets defined by IWSLT[1]. Each set consists of 8–11 single-speaker talks of around 10 minutes each. In most of our experiments we combined these three sets into a single test set containing 27 talks, comprising approximately 5 hours of speech. In all cases we used the manual speech/non-speech segmentations supplied for the IWSLT evaluation.

Our full acoustic training set consisted of 813 TED talks recorded prior to 2010, comprising 145 hours of speech segments in total. Transcriptions were obtained through a lightly supervised alignment procedure [39]. A language model was trained on transcribed TED talks, and the Europarl, News Crawl and Gigaword corpora [40]. This experimental setup is somewhat similar to the Kaldi TED-LIUM recipe[2], but is fully compliant with the rules for the IWSLT evaluation. Note also that the recipe uses only the `dev2010` and `tst2010` test sets.

A. Baseline systems

Using the Kaldi toolkit, an initial GMM system was trained on 13-dimensional MFCC features following a standard recipe, which involves several iterations of training data realignment, decision-tree building, and Gaussian mixing-up. The features for the final system used ± 4 frames of context, to which linear discriminant analysis (LDA), a maximum likelihood linear transform (MLLT), and a speaker-adaptive feature space (constrained) maximum likelihood linear regression (FMLLR) transforms were applied: a single FMLLR transform is estimated for each TED talk. We refer to the resulting features as FMLLR features. The baseline system trained on the full data set had approximately 10,000 tied states with 16 Gaussians per state.

Our DNN systems are trained on the FMLLR features derived from the GMM system. We used a fixed DNN architecture of 6 hidden layers with 2048 units per layer. The hidden layers used logistic sigmoid non-linearities, with a softmax function at the output layer which is discarded for decoding. All DNN weights were initialised using generative RBM pre-training [41], and fine-tuned with several epochs of cross-entropy training, using the targets and frame alignments from the GMM system. This training was performed on NVIDIA GPUs using an in-house tool based on the Theano library [42]. Stochastic gradient decent was used with a minibatch size of

[1] http://iwslt.org

[2] http://kaldi-asr.org/

256 samples. We selected a learning rate of 0.16 based on experiments on the full data set, and reduced the learning rate for successive epochs using the "newbob" schedule [43]. In all cases, 10% of the talks were held out for use as a validation set. DNN weights were exported for use in the Kaldi decoders.

When the DNNs are trained with MTL, we use an equal weight for each task: for example, when there are two tasks, we use $\lambda = 0.5$ in Equation 6. We have found that results are not sensitive to the value of λ. In practice, since our implementation presents tasks separately at the minibatch level, this weighting is achieved by using a halved learning rate, 0.08, for each task.

We also carried out experiments using sequence training [30] to optimise the sequential minimum Bayes risk (sMBR) criterion. Lattices for all training utterances were generated using the cross-entropy trained DNN, and two iterations of sequence training were performed using a learning rate of 10^{-5}. We did not tune this recipe, but found that it gave good results in practice.

At test time, a first-pass decoding with the GMM was used to generate alignments on which to estimate an FMLLR transform for each talk in the test set, following which the required FMLLR features were obtained for use in decoding with the DNNs. These features were fixed for all experiments. Decoding was carried out using a trigram LM, pruned with entropy 10^{-7}. Unless otherwise stated, all results are given with this LM, although in some cases we also rescore lattices with an unpruned 4-gram LM to demonstrate the best recognition accuracies.

B. Reduced training data setup

We performed extensive experiments to investigate the interaction between the quantity of training data and the effect of standard and multitask DNN training, as well as the effect of varying the number of tied states. To investigate the effects of varying the quantity of training data we defined 10%, 20%, 50% and 75% subsets of the full training data, corresponding to the approximate proportion of DNN training data retained, *after* removing 10% of the data to act as a validation data set, fixed for all data conditions. Although the percentages of data were calculated with respect to the accumulated length of speech segments, data selection was carried out at the whole talk level. Since there are many talks in the training set, and all are of similar duration, there is little difference in practice. The selection is carried out alphabetically by the name of the talker, so is effectively random with respect to the acoustic characteristics.

When investigating the effects of the quantity of training data and number and configuration of tied state targets, we aimed to control for the quality of alignments and input features – in addition to keeping the experimental configuration as simple as possible. Therefore we do not train GMMs from scratch on each reduced quantity of training data. We instead used the final full-data GMM system to derive the FMLLR features for each speaker to be used as inputs to the DNN; these are fixed for all experiments. Using state alignments from the full-data GMM, we generate a new decision tree based on occupancy statistics in the reduced data, fixing the desired number of leaves. This guarantees that the state-tying is appropriate for the smaller data set in terms of the number of samples per state. However, we do not realign the data, instead converting the original full-data alignments to use the new decision tree via a deterministic mapping of the logical triphones. This ensures that the quality of the targets used as DNN targets is consistent across experiments. The DNNs were initialised using RBM pre-training on the full data set.

Finally, when performing sequence training with the reduced training data, we used denominator lattices generated by the full-data models. Only the word alternatives from these lattices are retained, being used to generate a new lattice with a matched set of tied states, allowing sequence training of the reduced-data DNN. This avoids the significant computational overhead of regenerating the lattices from scratch for every possible decision tree and data quantity condition, while yielding the expected improvements in performance.

TABLE I
BASELINE RESULTS ON THE TED LECTURE TASK FROM SYSTEMS TRAINED ON THE FULL TRAINING DATA

System	Test set			mean
	dev2010	tst2010	tst2011	
ML GMM	21.9	20.6	17.4	20.3
+ 4gram	19.5	17.9	14.8	17.7
CE DNN	17.8	16.7	14.1	16.5
+ sequence	15.9	14.7	12.2	14.5
+ 4gram	13.4	12.0	10.0	12.0

IV. EXPERIMENTS

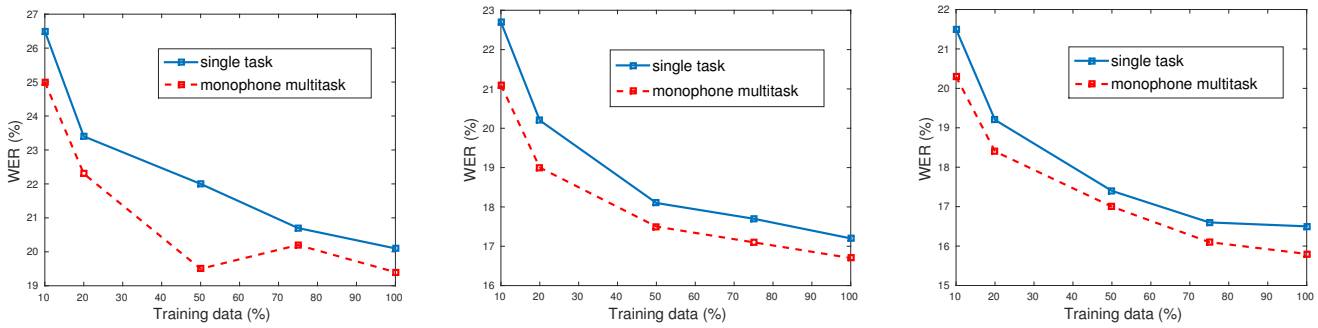
In Table I we present baseline results on the three test sets from the TED lecture task using systems trained on the full training set. All these systems use 8,000 tied states; alignments for cross-entropy DNN training are obtained using the baseline maximum-likelihood GMM. As expected, there are substantial gains from sequence training of DNNs on this task. Results are competitive with other state-of-the-art systems reported in the literature [38][3]. In further results, we give only WER averaged over all test sets (this is weighted by the number of words in each set) and do not rescore with the 4-gram LM.

We additionally investigated the effect of RBM pre-training on the performance of DNNs trained with cross-entropy. On the full training set, we found no significant difference in WER. However, experiments on 50% of the full training set, demonstrated improvements from pre-training. We opted to use pre-training on the full data in all the following experiments.

A. Varying quantities of data

Our first experiment investigates the use of monophone targets as a secondary task for DNN training. Following the standard Kaldi recipe, monophone units are dependent on the position within the word (beginning/end/internal/singleton). There are 186 such monophones in total. Figure 3 compares

[3]NB. Some of the systems reported at IWSLT were trained on additional corpora



(a) (b) (c)

Fig. 3. Comparing standard DNNs with the use of monophones as a secondary task, for different sizes of primary task: (a) 1000; (b) 4000; (c) 8000.

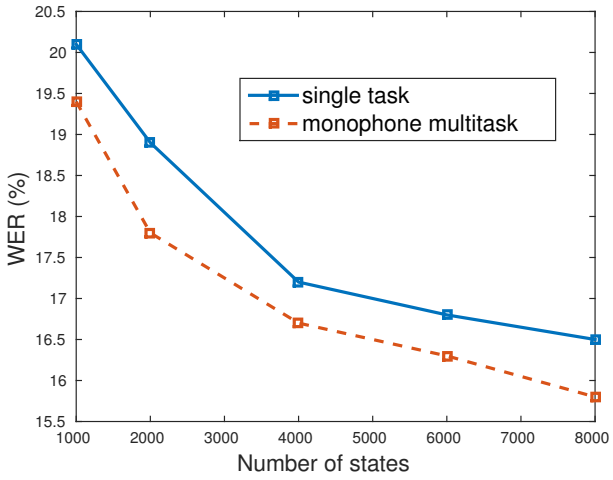
this multitask system with the standard single-task DNN, as the quantity of training data is reduced to 10% of the full quantity. Full figures are provided in Table II. The monophone multitask systems consistently outperform the baseline over all quantities of training data, and this effect holds over different output sizes for the secondary task.

Interestingly, the relative gains from multitask learning here vary little with the quantity of data, suggesting that the secondary task is not simply acting as a smoothing prior, compensating for data sparsity in the higher-dimensional primary task. In other words, the effect of the secondary task is not purely one of regularisation.

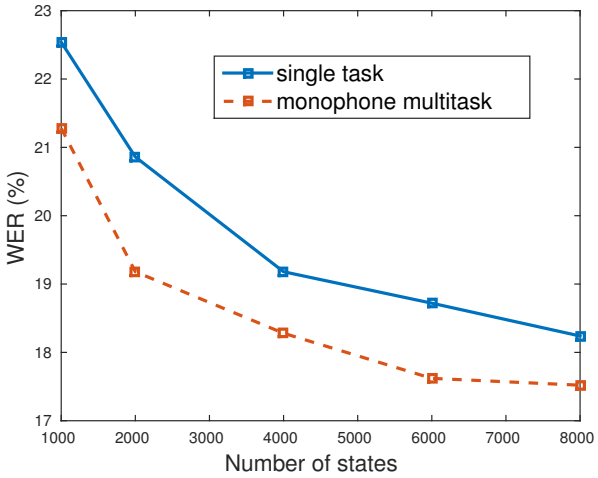
B. Varying the number of tied states

We also show the change in performance between single and multitask systems with a varying number of tied states for the primary task. Figure 4a shows this for models trained on the entire data set; Figure 4b shows an average at each size over all the subsets of the training data that we used. The full set of results can again be found in Table II. Once again, the results show remarkably consistent gains over the range of output sizes, giving further support to the suggestion that the effect is not merely one of smoothing the larger task.

As discussed in Section III-B, we controlled for the quality of the frame-level alignments across all data sizes through the use of a deterministic conversion between decision trees. This avoids the situation where the DNNs are trained on more noisy targets due to a poor quality alignment because of too many GMM states or too little data. This may help explain the consistency in WER reductions across the different conditions: we hypothesise that multitask learning is correcting for the inherent weakness of using a single set of clustered triphones as a target.



(a) Full data



(b) Averaged over all data sizes

Fig. 4. Comparing standard DNNs with the use of monophones as a secondary task with varying number of tied states.

C. Sequence training

As discussed in Section I, a possible explanation for the benefits of using a context-independent secondary task is that it encourages the network to learn a hidden representation that gives greater benefits towards discrimination between phones, rather than between triphones with the same central phone, unlike the standard cross-entropy criterion which treats all discriminations equally. Sequence training of DNNs may be viewed as an alternative method for improving over cross-entropy training, optimising an objective function more closely related to word accuracy. Therefore, we one may conjecture

TABLE II
COMPARING SINGLE TASK AND MULTITASK OUTPUT WITH A RANGE OF TRAINING DATA AND TIED STATES

# states	Quantity of training data (%)									
	10		20		50		75		100	
	Single	MT	Single	MT	Single	MT	Single	MT	Single	MT
1000	26.5	25.0	23.4	22.3	22.0	19.5	20.7	20.2	20.1	19.4
2000	25.1	22.3	21.3	20.0	19.8	17.7	19.2	18.1	18.9	17.8
4000	22.7	21.1	20.2	19.0	18.1	17.5	17.7	17.1	17.2	16.7
6000	22.2	20.6	19.9	18.4	17.6	16.3	17.1	16.5	16.8	16.3
8000	21.5	20.3	19.2	18.4	17.4	17.0	16.6	16.1	16.5	**15.8**

that the gains from using a monophone secondary task will diminish when sequence training is applied.

In this section, we present results following the application of sMBR sequence training to the models from the previous sections. The exact implementation of this was described in Section III-B. The full set of results is in Table III. Figure 5a compares systems with and without sequence training across all different data sizes with a fixed 8,000 tied states. Figure 5b compares single and multitask systems when sequence training is applied, with a varying number of tied states. Finally, Figure 5c illustrates the relative gains from sequence training in both standard and multitask systems. There are substantial gains from the use of sequence training – the relative gains are larger at larger data sizes. Interestingly, however, the gains are remarkably similar between the systems, so that use of the secondary task continues to provide consistent benefits over single task systems when used in combination with sequence training. The relative gains from sequence training appear slightly higher with the multitask systems at smaller data sizes, and slightly lower with larger data. Taken together with the results in the previous sections, this suggests that the observed benefits may not necessarily be entirely explained by either the small size of the secondary task or the fact that context-independent targets are used.

D. Alternative secondary tasks

We conducted further experiments to determine the effect of choosing alternate, larger secondary tasks. We found previously that factorising left and right phonetic context and using each as secondary tasks can also be effective [22]. Motivated somewhat by Xu et al [44], where multiple complementary decision trees were used in system combination, we elected to use differently-sized decision trees as the secondary tasks. For each different training data size, we re-used each of the decision trees with 1,000-6,000 leaves, generated for the primary tasks in the previous experiments, but this time in the role of a secondary task for a system with 8,000 tied states. This is a simple way of varying the secondary task size. Again, the tasks are deterministically related. It should be noted that the use of these secondary tasks increases the computational cost of training compared to the much smaller monophone task, although there is no difference in decoding since the secondary task is discarded as usual.

Figure 6 and Table IV compares the results with the monophone multitask systems and the single task baseline. The findings are interesting: we see that *all* secondary tasks give significant gains over the baseline. However, there are no consistent conclusions about which size of secondary task is best, although it seems generally to be the case that larger secondary tasks have better performance.

TABLE IV
ALTERNATE SIZES OF SECONDARY TASKS

Secondary #states	Quantity of training data (%)				
	10	20	50	75	100
Single task baseline	21.5	19.2	17.4	16.6	16.5
1000	20.9	18.4	17.0	16.3	16.1
2000	20.4	18.7	16.9	16.3	16.2
4000	20.5	**18.3**	16.9	**16.1**	15.9
6000	20.5	18.5	**16.7**	16.3	16.1
Monophone multitask	**20.3**	18.4	17.0	**16.1**	**15.8**

TABLE V
COMBINING MULTIPLE TASKS

Secondary #states	Quantity of training data (%)				
	10	20	50	75	100
Single task baseline	21.5	19.2	17.4	16.6	16.5
Best task	20.4	18.3	16.7	16.1	15.9
Mean over tasks	20.6	18.5	16.9	16.3	16.1
Combining tasks	20.5	18.6	17.3	16.4	16.3

It is clear from these results that the gains from using alternative outputs in a multitask fashion do not simply derive from the smaller size of task. However, the use of more than one set of senone outputs is clearly beneficial; and given that the monophone multitask system generally achieves performance close to the best across most data conditions, we suggest that these findings support the theory that a problem with tied triphone state targets is their inherent arbitrariness – explaining why improvements are seen with multiple sets of targets as well as with the well-defined monophone task.

E. Combining multiple secondary tasks

Given that were able to obtain performance improvements with each of the secondary tasks in the previous section, we next performed experiments combining *multiple* secondary tasks (i.e. using multiple output layers). There are several ways in which this could be implemented; we decided to use a method that would be most comparable to the systems with a single additional task. Given that a single task is presented at the minibatch level in our standard implementation, we wished to avoid more presentations of each data point within a single epoch, while ensuring that the primary task continues to receive the same weighting. In our proposed scheme, tasks are selected randomly for an update: the primary task receives a probability of 50%, whilst the remaining 50% probability

TABLE III
COMPARING SINGLE TASK AND MULTITASK DNNS FOLLOWING SMBR SEQUENCE TRAINING

# states	Quantity of training data (%)									
	10		20		50		75		100	
	Single	MT	Single	MT	Single	MT	Single	MT	Single	MT
1000	23.3	21.7	20.0	18.9	18.0	16.1	16.7	16.4	16.5	15.8
2000	23.2	19.8	18.5	17.7	16.7	15.2	16.2	15.4	15.8	15.2
4000	21.0	19.3	18.1	16.8	15.6	15.2	15.2	14.9	14.8	14.5
6000	20.8	19.1	18.3	16.6	15.3	14.4	15.1	14.6	14.6	14.4
8000	20.4	19.1	17.6	16.7	15.3	14.9	14.6	14.5	14.5	**14.0**

mass is shared evenly between secondary tasks. This means that we continue to use a halved learning rate of 0.08 for all tasks. An epoch ends when the primary task has presented the complete set of data.

The results are shown in the final row of Table V, where they are compared with best two-task systems from the previous section, and an average over all such systems. Whilst continuing to outperform the single task baseline, the system with multiple secondary tasks does not score better than the best two-task system in any data condition, or than the average score from the two-task systems. This seems to suggest that a further increase in output diversity does not help. However, it may be that, because less data is presented for each of the secondary tasks, the output layers for these tasks are less well estimated, potentially reducing the benefit of the error signals from these tasks in learning a good shared representation. We will investigate this further in future work.

F. Effects of task shuffling

Finally, we carried out experiments to investigate our intuition that MTL is most effective for correlated tasks when minibatches for each task are presented randomly, compared to a more standard implementation where both tasks are optimised jointly in every minibatch update. We trained variants of the monophone multitask models where both tasks are updated for the same data subset in successive minibatches. Owing to our system design, we were not able to share the forward pass for both tasks following Equation 5 exactly. Computing updates for each task over two minibatches is computationally less efficient, but we do not believe is otherwise materially different. Note that in both implementations, the data is randomised at the minibatch level.

The results are presented in Table VI. This confirms that our standard implementation with shuffled tasks is indeed consistently more effective than the alternative. Averaged over all data conditions, the shuffled implementation yields almost double the WER reduction compared to the other implementation.

G. Final results

Table VII present final results for the single best multitask system on the TED lecture task separated by test set. These systems use the full set of training data and have 8,000 states. Even in this largest data condition, and with the use of sequence training and a larger LM, relative gains of around 2.5% are observed.

TABLE VI
COMPARING IMPLEMENTATIONS OF MULTITASK LEARNING (SYSTEMS WITH 8000 STATES)

System	Quantity of training data (%)					mean
	10	20	50	75	100	
Single task baseline	21.5	19.2	17.4	16.6	16.5	18.2
Joint task update	20.8	18.5	17.0	16.4	16.4	17.8
Shuffled task update	20.3	18.4	17.0	16.1	15.8	17.5

TABLE VII
FINAL RESULTS ON THE TED LECTURE TASK

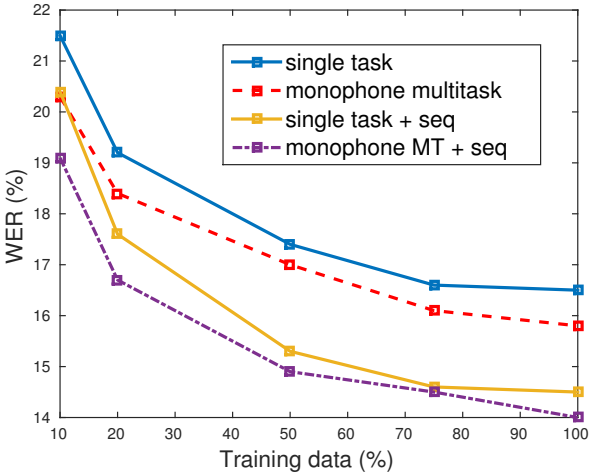
System	Test set			mean
	dev2010	tst2010	tst2011	
Single task system				
CE DNN	17.8	16.7	14.1	16.5
+ sequence	15.9	14.7	12.2	14.5
+ 4gram	13.4	12.0	10.0	12.0
Monophone multitask				
CE DNN	17.3	15.8	13.5	15.8
+ sequence	15.2	14.2	11.7	14.0
+ 4gram	**12.9**	**11.7**	**9.9**	**11.7**

V. CONCLUSION

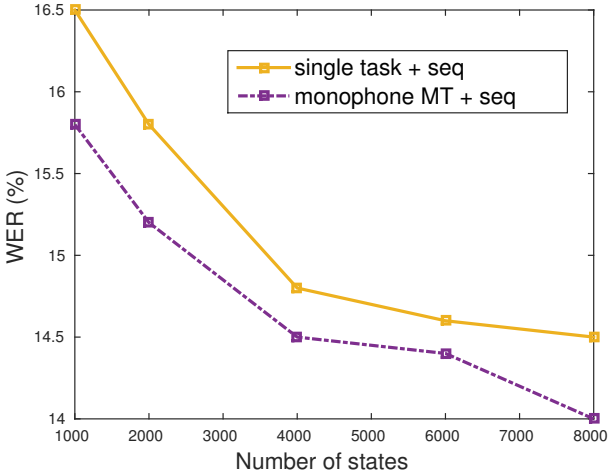
This paper has demonstrated improved performance in hybrid DNN systems for ASR from the use of multitask learning where secondary tasks either use context-independent targets or alternative state-tying schemes. Improvements are found across a range of data sizes and with varying numbers of tied states. The benefits appear to be greater when batches of data are presented independently for each task. There are generally relative gains of between 2-6% WER.

Multitask learning is an intriguing technique. In a sense, it is surprising that simply by training a context-dependent DNN to additionally predict secondary tasks – that contain no additional information about the data at all – we find consistent improvements over the standard training method. Yet we have also shown that the benefits are not simply due to a smaller secondary task acting as a lower dimensional prior to regularise the network when there is sparse training data; nor are they purely related to the weakness of the cross-entropy criterion when applied to tied-state targets, since we have found that benefits persist when sequence training is applied.

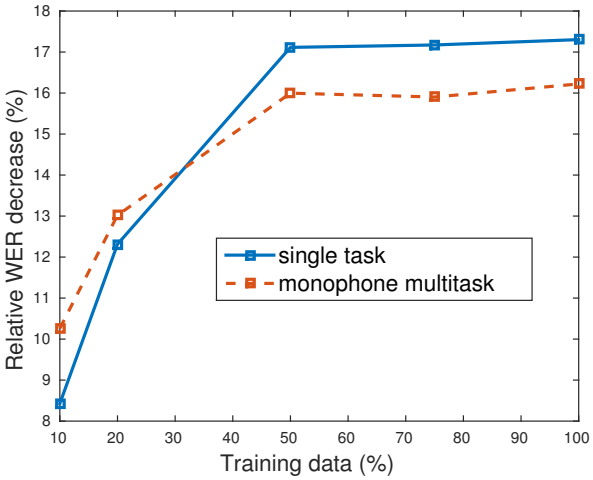
Further work is needed to analyse exactly which properties of the multitask technique are responsible for improvements in the primary DNN. The most plausible explanation arising from these experiments is that generating diversity in the rather arbitrary senone outputs avoids the model over-fitting to a single set of targets. However, increasing this diversity further



(a) Varying training data, 8000 states



(b) Varying number of states, full training data



(c) Relative gains from sequence training for single and monophone multitask systems

Fig. 5. Comparing standard and multitask models with sequence training

does not appear to help. Exploring exactly how this diversity may be optimally exploited will be the subject of future work.

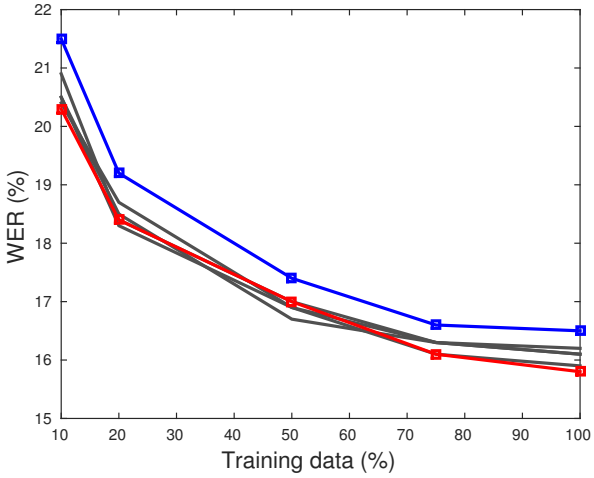


Fig. 6. Comparing alternative secondary tasks with varying quantity of training data. We show the single task system (blue), monophone multitask system (red) and systems with the alternative decision trees of different sizes (grey).

REFERENCES

[1] L. R. Bahl, R. Bakis, F. Jelinek, and R. L. Mercer, "Language-model / acoustic-channel-model balance mechanism," *IBM Technical Disclosure Bulletin*, vol. 23, no. 7B, pp. 3464–3465, December 1980.

[2] R Schwartz, Y Chow, O Kimball, S Roucos, M Krasner, and J Makhoul, "Context-dependent modeling for acoustic-phonetic recognition of continuous speech," in *Proc IEEE ICASSP*, 1985, vol. 10, pp. 1205–1208.

[3] K.F. Lee, *Automatic Speech Recognition: The development of the SPHINX system*, Kluwer Academic Publishers, 1989.

[4] S.J. Young, J.J. Odell, and P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the Workshop on Human Language Technology*, 1994, pp. 307–312.

[5] J. Baker, "The DRAGON system–an overview," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 24–29, 1975.

[6] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[7] M.Y. Hwang, X.D. Huang, and F.A. Alleva, "Predicting unseen triphones with senones," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 6, pp. 412–419, 1996.

[8] M. M. Hochberg, S. J. Renals, A. J. Robinson, and G. D. Cook, "Recent improvements to the ABBOT large vocabulary CSR system," in *Proc. IEEE ICASSP*, 1995, pp. 69–72.

[9] A. J. Robinson, G. D. Cook, D. P. W. Ellis, E. Fosler-Lussier, S. J. Renals, and D. A. G. Williams, "Connectionist speech recognition of broadcast news," *Speech Communication*, vol. 37, no. 1–2, pp. 27–45, 2002.

[10] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[11] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011.

[12] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[13] M. Bacchiani and D. Rybach, "Context dependent state tying for speech recognition using deep neural network acoustic models," in *Proc. ICASSP*, 2014, pp. 230–234.

[14] C. Zhang and P. C. Woodland, "Standalone training of context-dependent deep neural network acoustic models," in *Proc. ICASSP*, Florence, Italy, 2014.

[15] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent networks in continuous speech recognition," in *Automatic Speech and Speaker Recognition – Advanced Topics*, CH Lee, KK Paliwal, and FK Soong, Eds., pp. 233–258. Kluwer Academic Publishers, 1996.

[16] A Graves, A-r Mohamed, and G Hinton, "Speech recognition with deep recurrent neural networks," in *Proc IEEE ICASSP*, 2013, pp. 6645–6649.

[17] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[18] H. Sak, K. Senior, A. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Proc. ICASSP*, 2015.

[19] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: a context dependent neural network for continuous speech recognition," in *Proc. IEEE ICASSP*, 1992, vol. 2, pp. 349–352.

[20] G. Wang and K.C. Sim, "Regression-based context-dependent modelling of deep neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 11, pp. 1660–1669, nov 2014.

[21] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," in *Proc. ICASSP*, 2015.

[22] P. Bell and S. Renals, "Complementary tasks for context-dependent deep neural network acoustic models," in *Proc. Interspeech*, 2015.

[23] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.

[24] O. Siohan and D. Rybach, "Multitask learning and system combination for automatic speech recognition," in *Proc. ASRU*, 2015.

[25] Z. Tüske, R. Schlüter, and H. Ney, "Multi-lingual hierarchical MRASTA features for ASR," in *Proc. Interspeech*, 2013.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009.

[28] Y.S. Abu-Mostafa, "Learning from hints in neural networks," *Journal of Complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[29] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. IEEE ICASSP*, 2009, pp. 3761–3764.

[30] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013.

[31] S. Parveen and P. Green, "Multitask learning in connectionist robust ASR using recurrent neural networks," in *Proc. Interspeech*, 2003.

[32] J. Stadermann, W. Koska, and G. Rigoll, "Multi-task learning strategies for a recurrent neural net in a hybrid tied-posteriors acoustic model.," in *Proc Interspeech*, 2005, pp. 2993–2996.

[33] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc. ICASSP*, 2013.

[34] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, "Joint acoustic modelling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *Proc. ICASSP*, 2014.

[35] C. Zhang and P.C. Woodland, "Context independent discriminative pre-training," Unpublished work, 2015.

[36] D. Chen and B. Mak, "Distinct triphone acoustic modelling using deep neural networks," in *Proc. Interspeech*, 2015.

[37] O. Siohan, "Sequence training of multi-task acoustic models using meta-state labels," in *Proc. ICASSP*, 2016.

[38] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Frederico, "Report on the 10th IWSLT evaluation campaign," in *Proc. International Workshop on Spoken Language Translation*, Heidelberg, Germany, December 2013.

[39] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. SLT*, 2012.

[40] P. Bell, F. McInnes, S. R. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN english ASR system for the IWSLT 2013 evaluation," in *Proc. International Workshop on Spoken Language Translation*, 2013.

[41] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[42] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. SciPy*, June 2010.

[43] S. Renals, N. Morgan, M. Cohen, and H. Franco, "Connectionist probability estimation in the DECIPHER speech recognition system," in *Proc. IEEE ICASSP*, 1992.

[44] H. Xu, G. Chen, D. Povey, and S. Khudanpur, "Modeling phonetic context with non-random forests for speech recognition," in *Proc. Interspeech*, 2015.

Peter Bell is a senior research associate at the University of Edinburgh. He received a BA in Mathematics from the University of Cambridge and a PhD in automatic speech recognition from Edinburgh. His research interests include domain adaptation, regularisation, and low-resource methods for acoustic modelling. He has an extensive portfolio of industrial collaborations.

Pawel Swietojanski holds a Ph.D. degree in Computer Science from University of Edinburgh, UK. His main research interests are in machine learning and its applications to speech processing, with a particular focus on learning representations for acoustic modelling in speech and speaker recognition. He currently works as a research scientist for Emotech Ltd. where he develops machine learning techniques for personal and social robots.

Steve Renals (M'91 — SM'11 – F'14) is professor of speech technology at the University of Edinburgh. He received a BSc from the University of Sheffield and an MSc and PhD from Edinburgh. He has previously had positions at ICSI Berkeley, the University of Cambridge, and the University of Sheffield. His research interests are in speech and language processing.