# Complementary tasks for context-dependent deep neural network acoustic models

*Peter Bell, Steve Renals*

Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH8 9AB, UK
{peter.bell, s.renals}@ed.ac.uk

## Abstract

We have previously found that context-dependent DNN models for automatic speech recognition can be improved with the use of monophone targets as a secondary task for the network. This paper asks whether the improvements derive from the regularising effect of having a much small number of monophone outputs – compared to the typical number of tied states – or from the use of targets that are not tied to an arbitrary state-clustering. We investigate the use of factorised targets for left and right context, and targets motivated by articulatory properties of the phonemes. We present results on a large-vocabulary lecture recognition task. Although the regularising effect of monophones seems to be important, all schemes give substantial improvements over the baseline single task system, even though the cardinality of the outputs is relatively high.

**Index Terms**: deep neural networks, multitask learning, context modelling

## 1. Introduction

Since the work of [1, 2], the use of clustered context-dependent (CD) phone state targets has been standard when using deep neural networks (DNNs) in a hybrid HMM configuration – that is, using the DNN to directly generate state likelihoods for use in an HMM-based decoder. The tied-state context-dependent units (also known as *senones*) are typically derived from a clustering obtained from a baseline GMM system, following the standard approach of [3], although DNNs have recently been used for this purpose [4, 5]. This state-tying is clearly important to avoid data sparsity issues due to the very large number of possible CD units.

Modelling tied-state CD units works well in practice, and authors have found the performance insensitive to the number of units [6, 7], in part due to the efficient sharing of parameters across outputs in the deep-structured model . Modelling CD units essentially yields a more powerful model that adjusts the model of each phone to take account of its local acoustic context. However, we have argued that a disadvantage of this approach is that when models are trained using the standard cross-entropy criterion, there is no distinction between tied states of the same monophone and tied states of different monophones: the DNN discriminates between each equally. Whilst discrimination between monophones has direct benefit in decoding, discriminating between two senones of the same monophone may yield only indirect benefits, and is determined only by the somewhat arbitrary choice of state clustering. This problem was recognised in early work in neural network acoustic modelling [8, 9, 10, 11, 12], where efforts were made to factorise the context-dependent probabilities and model them with independent nets. Given data $x_t$, the probability of phone state $q_k$ in left-context $c_j^l$ and right context $c_l^r$ is modelled according to:

$$p(q_k, c_j^l, c_l^r|x_t) = p(q_k|x_t)p(c_l^r|q_k, x_t)p(c_j^l|q_k, c_l^r, x_t) \quad (1)$$

where $k$ indexes the set of phones and $j, l$ index the set of contexts. Whilst this expression is exact, its severe limitation is that the both phone and context labels from the decoding hypothesis are used as inputs to the networks, rendering intractable the forward pass computation during decoding. The solution proposed in [9] – factorising the nets into separate structures for each input, combining only at the final sigmoid layer – effectively models the effect of context independently of the phone target, a significant reduction in modelling power.

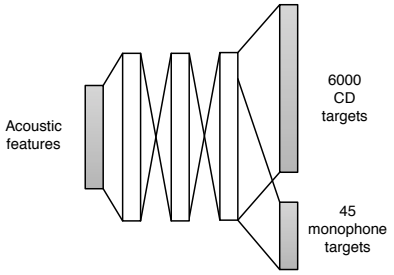


Figure 1: An example multitask network structure

In our recent work [13] we proposed to use a multitask DNN, illustrated in Figure 1 to jointly model tied-state CD targets and the context-independent (CI) monophone targets. Multitask learning [14], in which a shared network structure is used to learn complementary tasks, is known to improve generalisation by learning a better shared hidden representation for both tasks. In this case, the use of monophone targets as a secondary task aims to add weight to the form of discrimination that is more relevant for decoding, without requiring the explicit factorisation of Equation 1. It can also be seen as a smoothing term based on a lower dimensional, simpler task. We found that this structure gives substantial benefits – particularly when training data is limited – over standard DNNs with generative pre-training, and also DNNs initialised by training on monophone targets.

In this paper we seek to better understand the benefits of multitask learning in this setting. We question whether the observed improvements derive from the use of a low-dimensional

This research was supported by EPSRC Programme Grant grant, no. EP/I031022/1, Natural Speech Technology (NST). Information on data used in this research may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786

secondary task reducing noise in the training signals; whether the gains come from explicitly discriminating between monophones; or whether the disadvantage of modelling tied-state CD units is the arbitrariness of the labels used as targets for the DNNs. To do this, we extend the framework to model alternative context-related tasks of varying dimensions, generating alternative clustering schemes according to broad articulatory properties of the context, explained in the following section.

## 2. Multitask for CD modelling

Multitask learning [14] is now a well-known technique in DNN-based acoustic modelling. It found early application to noisy ASR in [15]. More recently, it has been used for joint modelling of triphone and trigrapheme units in [16], and for text-to-speech [17]. Most closely related to the current work is [18], where a variety of complementary tasks of small size were investigated to enhance context modelling of a monophone DNN. However, this work differs substantially in that [18] conducted experiments on the TIMIT corpus where the small quantities of training data mean that monophone modelling is standard; in contrast, we use a large-vocabulary ASR system trained on 10s or 100s of hours of speech, we always use a large set of tied state triphones as the primary task and can model much larger related tasks.

We consider a task $A$ as being a mapping from a set of $T$ training frames to a set of labels, that is:

$$\begin{aligned} A : \{t : 1 \leq t \leq T\} &\rightarrow \{1, \ldots, L\} \\ t &\mapsto y_t^A \end{aligned} \tag{2}$$

We denote the cardinality of task $A$ as $L^A$. In what follows, we use $A$ to indicate the primary task – in the sense that these will be the outputs of the network used at test time. We can define the objective for task $A$ by the negative cross-entropy

$$\mathcal{F}_A(\theta) = \sum_t \log p(y_t^A | x_t; \theta) \tag{3}$$

In learning task $A$, we maximise this objective function with respect to the parameters $\theta$. The optimisation is carried out with stochastic gradient descent, computing the gradient with respect to small mini-batches of training data, updating the parameters with a small step size, and iterating over the complete set for multiple epochs until the the convergence is reached on held-out validation data. .

In multitask learning, we define an additional task $B$, generating a new labelling for the training frames, and obtain the objective $\mathcal{F}_B(\theta)$. We alternate updates of the parameters according to the objectives $\mathcal{F}_A(\theta)$ and $\mathcal{F}_B(\theta)$ at the minibatch level . In a typical configuration, all parameters except those in the output layers are shared, although it is possible for parameters in lower layers to influence only one task. An alternative would be to optimise a joint objective $\lambda\mathcal{F}_A(\theta) + (1-\lambda)\mathcal{F}_B(\theta)$, which would combine both tasks within a single minibatch.

Multitask learning is closely related to other methods proposed for improving neural network generalisation: for example, in curriculum learning [19] the network is trained on initially on a simple task $B$ (in the sense of the distribution of $y_t^B$ having low entropy), moving to the primary task $A$ as the training proceeds. $\mathcal{F}_B(\theta)$ could also be viewed as KL-divergence regularisation term in the manner of [20], with the effect of moving $p(y|x_t;\theta)$ closer to the empirical distribution defined by $y_t^B$.

[14] postulates a number of reasons for the observed success of multitask learning. One is that the use of additional tasks minimises the effect of noise in the error signals used in backpropagation since they are effectively averaged across tasks. We would expect this effect to be more pronounced when the secondary task has low cardinality $L^B$. A second is the notion of *eavesdropping* – a hidden representation may may be easily learnt for $B$ but not for $A$, so sharing tasks allows $A$ to eavesdrop in the information from $B$. Specific to our problem setting, as discussed in Section 1 is the issue that the primary task may not necessarily be the ideal target to optimise, so including diverse alternative targets may be beneficial. We have explored three auxiliary tasks for the multitask learning of CD models: context-independent phone modelling, phone context modelling, and articulatory context modelling.

### 2.1. Context-independent tasks

The task we investigated previously was the predicting monophone units [13]. Monophone targets have previously been used as an initialisation of DNNs prior CD training [21], a form of curriculum learning. In this work, for consistency with the other tasks investigated, we instead use *monophone state* targets $y_t^{\text{ms}} = q_t$. The motivation for using context-independent targets is twofold: firstly, the task has small size ($L^{\text{ms}} = 45 \times 3 = 135$) so can be expected to have a strong regularising effect. Secondly, we expect that biasing the DNN towards monophone states encourages discrimination more beneficial for recognition, over discrimination between different tied states of the same monophone.

### 2.2. Phone context tasks

As the converse of using context-independent targets, we also investigate using tasks related to the precise context, separating by left and right context. By incorporating context explicitly into the DNN outputs, we expect to achieve better modelling of context-related effects in the input features, helped by the use of wide frame context in the inputs, as is standard for hybrid DNN systems. The two tasks are respectively defined by the tuples of left and right phone contexts along with the target phone state:

$$\begin{aligned} y_t^{\text{lc}} &= (c^l, q_t) \\ y_t^{\text{rc}} &= (c^r, q_t) \end{aligned} \tag{4}$$

Note that we model the effects of context on specific target CI phone state, in contrast to the approach of [18]. This results in task sizes $L^{\text{lc}} = L^{\text{rc}} = 5943$, similar to the size of the primary CD task. This is better matched to the standard clustering approach in large-vocabulary ASR.

### 2.3. Articulatory context tasks

The method of detecting articulatory features in speech has been well-studied [22, 23]. Rather than considering phonemes as atomic units, an articulatory-based classification scheme characterises a phoneme according to a series of articulatory categories: place and manner of articulation, voicedness and so on. Our use of these features to define context tasks is motivated by the recent work of [24], where single neural networks were used to predict CD units clustered according to a range of articulatory features and later combined with a regression model to generate probabilities for unseen triphones.

The opportunity to factorise the multiple articulatory properties of each phoneme across tasks represents an attractive application of multitask learning, allowing data sharing between

Table 1: *Articulatory features*

| Feature | Categories |
|---|---|
| Place | front vowel, central vowel, back vowel, coronal, palatal, labial, velar |
| Manner | high vowel, mid vowel, low vowel, fricative, nasal, stop cons, approximant |
| Voicedness | voiced, unvoiced |
| Miscellaneous | short vowel, long vowel, diphthong, retroflex, affricate, alveolar, constituent, non-constituent |

different context phonemes. It has previously found to be beneficial in modelling CI units on the TIMIT database [25]. We adopt a similar classification scheme to that in [24], summarised in Table 1. The silence phoneme has its own category in all tasks.

For each feature, we define a left and right context task analogously to the phoneme context tasks in Section 2.2, giving eight tasks in all. If $f_i(c),\ i \in \{1, \dots 4\}$ defines the $i^{\text{th}}$ feature for phoneme $c$ , the tasks for feature $i$ are given by the tuples $(f_i(c^l), q_t)$, $(f_i(c^r), q_t)$. This scheme has two potential benefits: firstly, through the factorisation of phonemes into features we reduce the maximum cardinality of a task to 1455; secondly, we have effectively generated eight different context-clustering schemes, rather than the single scheme used in standard triphone clustering, allowing us to investigate whether using the fixed arbitrary clustering is indeed damaging to performance, as hypothesised.

# 3. Methodology

## 3.1. ASR task

As in previous work, we carried out experiments on the development sets of the TED English transcription task from the IWSLT Evaluation [26]. The data consists of single-speaker talks of around 10 minutes' duration. We present results on 27 pre-segmented talks, combining the `dev2010`, `tst2010` and `tst2011` sets. All experiments use a trigram language model trained on a corpus of TED talks, and the Europarl, News Crawl and Gigaword corpora, following the IWSLT rules.

For acoustic training data, we used subsets of our full training corpus of 813 TED talks, yielding 131 hours of speech segments after holding out validation data. We also investigated the performance of models trained on data totalling 65 hours of speech segments and a 26 hours, respectively 50% and 20% of the complete training set. The reduced training set sizes were chosen partly to reduce experiment turnaround time, and partly to better highlight the difference between methods, since we previously observed that the benefits of multitask learning diminish as the quantity of training data increases [13]. The use of two sizes of training set allows us compare the robustness to limited training data of the different tasks investigated. The validation data was fixed for all sizes of training set.

## 3.2. Baseline system

As acoustic features, we used 13 perceptual linear prediction (PLP) coefficients with first and second order deltas. An monophone HMM-GMM system trained on these features was used to obtain a state-clustering of context-dependent phone units, and subsequently a GMM modelling these tied states. We used approximately 6,000 tied states with 16 Gaussians per state. The final GMM system was used to assign a frame-level phone alignment of the training data – which was fixed in all DNN experiments – and also to estimate a single constrained maximum likelihood linear regression (CMLLR) transform for each training speaker, which were used to generate speaker-normalised input features for the DNNs. At test time, equivalent transformations were estimated for each speaker using a first-pass decode with the speaker-independent GMM system.

All DNNs were trained to generate posterior probabilities over the labels for the target tasks by using backpropagation over the full structure to optimise the framewise cross-entropy criterion shown in Equation 3 with respect to the relevant task. In the baseline case, the DNN was used to purely to generate posterior probabilities over the 6,000 tied state targets. In all subsequent experiments, we fixed the primary task to be the same as the baseline, so that all multitask DNNs had one of the task predicting the 6,000 posterior probabilities. These outputs were the only ones used at test time, scaled by state prior probabilities to produce pseudo log likelihoods for use in hybrid-HMM decoding.

All DNNs used 6 hidden layers with 2048 units per layer. The hidden layers use logistic sigmoid non-linearities, with a softmax function at the output layer. For input features, we used the speaker-normalised PLP features in a 9-frame window ($\pm 4$ frames of context). Training was performed on NVIDIA GeForce GTX 690 GPUs using an in-house tool based on the Theano library [27].

In our previous work [13] we carried out experiments to determine the gain from multitask training with different DNN initialisation schemes. We found that there were generally no consistent benefits to using either generative pre-training with RBMs [28] or monophone pre-training [21]. In this work, to reduce the set of experimental combinations, we elected to use a simple random initialisation.

## 3.3. Multitask DNN training

Multitask DNNs were trained in a similar manner to the baseline single-task DNNs. In our implementation, task updates with stochastic gradient descent are alternated at the mini-batch level. When there are more than two tasks, updates for tasks are performed in rotation. Except for the output layer weights, all parameters are shared between all tasks, and every iteration of backpropagation updates all shared parameters. Training data is shuffled differently for each task, so that successive updates do not cause implicit learning of between-task correlations.

The learning rate is of course, an important tuning parameter. Given the number of configurations and tasks, it was not possible to exhaustively optimise the learning rates for each task independently. However, we had previously experimentally optimised the learning rate schedules for single-task DNNs: to begin with, we applied the same recipe, obtaining the learning rate for a single task by dividing the single-task learning rate by the number of tasks, thus keeping the effective learning rate over a complete epoch the same as the baseline. A problem with this approach is that it reduces the weight of the primary task as the number of tasks increases; we therefore investigated an alternative where the learning rate of the primary task was fixed at half the single task rate, sharing the remaining weight between the secondary tasks. In this scheme the updates from the primary task are the same in all cases as when there are two tasks.

In all experiments, learning rates were reduced independently for each task using the "newbob" schedule, measuring

the performance of the net on the task in question using frame error rate on the validation data.

## 4. Results

We present results on systems trained with the complementary tasks described in Section 2, comparing the performance with a standard single-task DNN trained purely to predict CD targets. Table 2 summarises the task configurations: "ms" indicates the task of predicting monophone states; "lc/rc" is the two tasks of predicting the monophone state jointly with the left or right phonetic context; "AF" denotes the task where the left or right context is decomposed according to four articulatory features.

Figure 2 shows the change in performance of baseline and multitask methods with varying quantities of training data. We observe that the same trends reported in our previous work continue to hold: all multitask methods outperform the single task baseline. As expected, the effect is most pronounced when the quantity of training data is smaller, confirming that the use of multitask learning has the effect of improving the generalisation of the DNN. More interestingly, this effect occurs not only in the situation where the secondary task has a small size – the same trends are observed when the secondary tasks are almost as large as the primary CD task. This suggests that the regularising effect is not simply due to the secondary task acting as a low-dimensional prior.

We give more detailed experimental results in Table 3. In addition to the three standard multitask setups, we also investigated adding the smaller monophone state task as an addition to the context related tasks (labelled "+ms"). Systems with the task of predicting state clustered units based only on either the left or right contexts performs similarly to those using the monophone state task. This implies that the role of the secondary tasks in preventing over-fitting of the DNN to CD targets from a single state clustering is an important one. However, in the smallest data case, there were additional benefits to adding the monophone state task as a fourth task for the DNN, which may be due to the smoothing effect of the smaller task.

In the final section of the table, the use of multiple alternative left/right clusterings based on articulatory features also results in substantial improvements over the baseline, supporting the conclusions above. However, these systems do not generally perform as well as the other two schemes. We speculated that this may be because the much larger total number of tasks prevents good convergence on the primary task: the decrease in performance caused by further adding the monophone state task would seem to support this. To investigate this theory, we carried out further experiments, assigning half the single task learning rate to the primary CD task as discussed in Section 3.3. This reduced WER from 18.1% to 17.3% and from 16.1% to 15.8% on the 26-hour and 65-hour training sets respectively, supporting the hypothesis.

Table 2: *Multitask configurations*

| System | # Tasks | Task cardinalities |
|---|---|---|
| Baseline CD | 1 | 6000 |
| Multi ms | 2 | 6000, 135 |
| Multi lc/rc | 3 | 6000, 5943×2 |
| +ms | 4 | +135 |
| Multi AF | 9 | 6000, 1059×2, 1059×2, 399×2, 1455×2 |
| +ms | 10 | +135 |

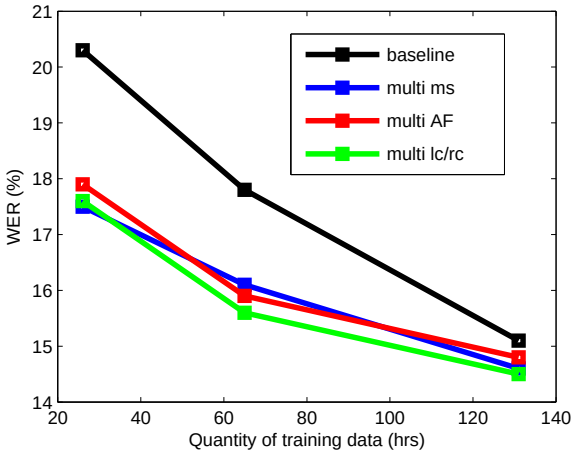


Figure 2: Results with varying quantities of training data

Table 3: *Results of multitask systems (WER%) on the TED task for varying quantities of training data*

| System | 26 hrs | 65 hrs | 131 hrs |
|---|---|---|---|
| Baseline DNN | 20.3 | 17.8 | 15.1 |
| Multitask ms | 17.5 | 16.1 | 14.6 |
| Multitask lc/rc | 17.6 | **15.6** | **14.5** |
| + ms | **17.3** | 15.7 | - |
| Multitask AF | 17.9 | 15.9 | 14.8 |
| + ms | 18.1 | 16.1 | - |

## 5. Conclusions

We have shown that multitask learning with alternative context-related targets is an effective method for improving the performance of hybrid DNN large-vocabulary ASR systems. The benefits are most pronounced when training data is scarce, but continue to hold for relatively large quantities. Further, we have shown that an important part of the effect is the manner in which multitask learning can reduce over-fitting to a single clustering of context dependent units used as targets in DNN training. We found that factorising the full triphone into left and right context tasks is as effective as using a single secondary task based on monophones – with both outperforming a standard DNN system. Both schemes help the DNN avoid learning discriminations which may not be beneficial when used in decoding.

We also found that splitting triphone contexts according to articulatory features outperforms a standard single task DNN. When more tasks are used, choosing appropriate learning rates becomes more difficult, and more work is needed. In future work, we intend to use combine the outputs from multiple tasks in a single decoding framework, hopefully allowing us to generate more accurate likelihoods over untied triphones, including those unseen or rarely seen in training data. The use of multiple tasks in our proposed framework also suggests the option to retrain or adapt the DNNs using specific tasks, perhaps selected according to the quantity of data available for this purpose.

## 6. Acknowledgements

We would like to thank Pawel Swietojanski for the use of his DNN training tools, and for many interesting discussions.

# 7. References


[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMS," in *Proc. ICASSP*, 2011.

[2] 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.

[3] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Computer Speech & Language*, vol. 8, no. 4, pp. 369–383, 1994.

[4] 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.

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

[6] G. Li, H. Zhu, G. Cheng, K. Thambiratnam, B. Chitsaz, D. Yu, and F. Seide, "Context-dependent deep neural networks for audio indexing of real-life data," in *Proc.SLT*, dec 2012.

[7] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription," in *Proc. ASRU*, 2013.

[8] N. Morgan and H. Bourlard, "Factoring networks by a statistical method," *Neural Computation*, vol. 4, no. 6, pp. 835–838, 1992.

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

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

[11] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent connectionist probability estimation in a hybrid HMM-neural net speech recognition system," *Computer Speech and Language*, vol. 8, pp. 211–222, 1994.

[12] D. Kershaw, T. Robinson, and S. Renals, "The 1995 ABBOT LVCSR system for multiple unknown microphones," in *Proc. ICSLP*, 1996, pp. 1325–1328.

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

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

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

[16] 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.

[17] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck featues for speech synthesis," in *Proc. ICASSP*, 2015, to appear.

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

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

[20] 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 *Proc ICASSP*, 2013.

[21] C. Zhang and P. Woodland, "Context independent discriminative pre-training," unpublished work.

[22] J. Frankel and S. King, "A hybrid ANN/DBN approach to articulatory feature recognition," in *Proc. Eurospeech*, 2005.

[23] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 333–353, 2000.

[24] G. Wang and K. 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.

[25] R. Rasipuram and M. Magimai-Doss, "Improving articulatory feature and phoneme recognition using multitask learning," in *Proc. ICANN*, ser. LNCS 6791, H. et al, Ed. Springer-Verlag, 2011.

[26] 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.

[27] 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*, Jun. 2010.

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