CONTEXT-DEPENDENT MODELLING OF DEEP NEURAL NETWORK USING LOGISTIC REGRESSION

Guangsen Wang, Khe Chai Sim

Computer Science Department, School of Computing, National University of Singapore

ABSTRACT

The data sparsity problem of context-dependent acoustic modelling in automatic speech recognition is addressed by using the decision tree state clusters as the training targets in the standard context-dependent (CD) deep neural network (DNN) systems. As a result, the CD states within a cluster cannot be distinguished during decoding. This problem, referred to as the *clustering* problem, is not explicitly addressed in the current literature. In this paper, we formulate the CD DNN as an instance of the canonical state modelling technique based on a set of broad phone classes to address both the data sparsity and the clustering problems. The triphone is clustered into multiple sets of shorter biphones using broad phone contexts to address the data sparsity issue. A DNN is trained to discriminate the biphones within each set. The canonical states are represented by the concatenated log posteriors of all the broad phone DNNs. Logistic regression is used to transform the canonical states into the triphone state output probability. Clustering of the regression parameters is used to reduce model complexity while still achieving unique acoustic scores for all possible triphones. The experimental results on a broadcast news transcription task reveal that the proposed regression-based CD DNN significantly outperforms the standard CD DNN. The best system provides a 2.7% absolute WER reduction compared to the best standard CD DNN system.

Index Terms— Context-Dependent Modelling, Deep Neural Network, Logistic Regression, Canonical State Modelling, Articulatory Features

1. INTRODUCTION

Context dependent (CD) modelling is widely adopted in the large vocabulary continuous speech recognition (LVCSR) systems to handle the co-articulation effects. Context dependency raises an important issue: the number of CD phones grows exponentially with the width of the context. In addition, a considerable number of them have limited number of occurrences or even unseen in the training corpus. To address this data sparsity problem, parameter sharing [1] is used. However, this leads to the *clustering* problem where all the CD phone states within a cluster share the same set of parameters, making them indistinguishable during decoding. Directly predicting all the CD state posteriors in the hybrid Neural Network/Hidden Markov Model (NN/HMM) system is not practical as it leads to a NN with a huge number of outputs. Both efficient computation and robust estimation of the model parameters will become issues. Thus, factorisation to smaller networks based on conditional probabilities is usually applied to circumvent this problem [2, 3, 4, 5].

Over the past few years, the development of machine learning algorithms [6, 7] and general-purpose graphics processing unit (GPGPUs) have made possible the training of Deep Neural Networks (DNNs), which are essentially NNs with many hidden layers (≥ 5). With proper pre-training, DNNs can accommodate thousands of output units. The standard CD modelling technique for DNNs is to use the decision tree state clusters [1] as the output targets in order to handle the data sparsity problem [8, 9, 10, 11]. However, the clustering problem is not explicitly addressed. Although it has been shown that CD DNNs can significantly outperform the discriminatively trained GMM/HMM and context-independent (CI) DNN over a variety of tasks from phone recognition [12] to LVCSR [8, 9, 10, 11], we believe that proper handling of the clustering problem will further improve the CD DNN performance.

In this paper, motivated by the canonical state modelling (CSM) [13] technique, a regression-based CD DNN modelling approach is proposed. Multiple sets of state clusters are used to represent the canonical states. Unlike the typical triphone state clusters, each set divides all the CD states into simpler disjoint clusters, which are easier to model, circumventing the data sparsity problem. These clusters are obtained based on the broad phone contexts, which are defined according to the articulatory features. DNNs are used to obtain the posterior probabilities of the broad phone state clusters. A logistic regression function is then used to transform the canonical states into the final state output probabilities. In addition, regression parameter tying is performed to reduce the model complexity. More interestingly, by carefully designing the broad phone state clusters such that each CD state can be uniquely identified using the canonical state representation, the resulting regression-based CD DNN is able to model each CD state distinctly, yielding a better context resolution compared to the conventional state clustering approach.

The remainder of the paper is organised as follows. Section 2 introduces the framework for the proposed regression-based context-dependent modelling. Section 4 describes ways to estimate the regression parameters. Section 3 defines the front-end detectors based on the broad phone categories. Experimental results are presented in Section 5. Section 6 summarises the findings and concludes the paper.

2. REGRESSION-BASED CONTEXT-DEPENDENT MODELLING

Context-dependent acoustic modelling is very important in automatic speech recognition systems for handling the *co-articulation* effects in continuous speech. However, even with only one phone context on each side, it is impractical to directly model the large number of triphones. Typically, parameter tying approaches, such as decision tree state clustering [1], are used to reduce the model complexity. The main drawback of such a *hard* clustering approach is the indistinguishability of the states within a cluster. To address this problem, this paper proposes a regression-based context-dependent model for DNN, where each state is *uniquely* defined given some regression bases. The proposed model is similar to Canonical State Modelling (CSM) [13] for CD GMM/HMM systems and Subspace

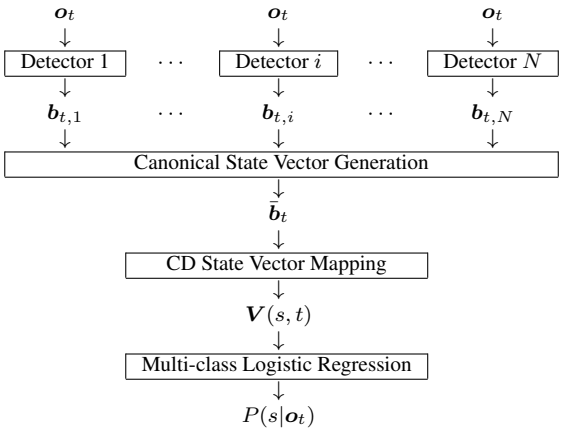

Fig. 1. *A schematic digram of the regression-based CD DNN*

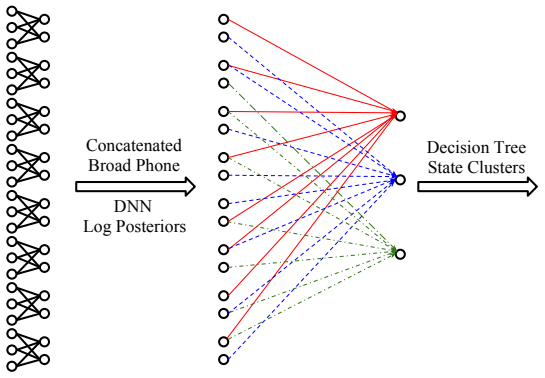

Fig. 2. *Sparse weight connection of the 2-layer regression NN*

GMM (SGMM) [17], where the CD states are the transformed versions of one or more canonical states. For the regression-based CD DNN, the canonical states are represented by multiple sets of state clusters and the CD state output probabilities are modelled as a regression of the log posterior probabilities of the state clusters. Figure 1 shows the structure of the regression-based CD model for DNN. There are three main components:

1. ***Canonical state vector generation:***
   For a given input feature vector, $\boldsymbol{o}_t$, N global detectors are used to predict a canonical state vector:

$$\bar{\boldsymbol{b}}_t = \left[\begin{array}{cccc} \boldsymbol{b}_{t,1}^\top & \boldsymbol{b}_{t,2}^\top & \ldots & \boldsymbol{b}_{t,N}^\top \end{array} \right]^\top$$

   where $\boldsymbol{b}_{t,i}$ denotes the posterior probability output of the ith detector. Each detector is used to predict a different set of state clusters. In this work, DNN detectors are used to predict biphone clusters using different categories of broad phone contexts. More details are given in Section 3.

2. ***CD state vector mapping:***
   For each CD state, s, a state descriptor $\boldsymbol{D}_s$ is used to map $\bar{\boldsymbol{b}}_t$ to a low dimensional CD state vector, $\boldsymbol{V}(s,t)$:

$$\boldsymbol{V}(s,t)[i] = \boldsymbol{b}_{t,i}[\boldsymbol{D}_s[i]]$$

   where $[i]$ indicates the ith vector element. $\boldsymbol{D}_s$ is an N-dimensional vector whose elements are the state cluster indicator for each detector. In this work, the detectors are designed so that all the CD states can be uniquely described by $\boldsymbol{D}_s$. See Section 3 for more details.

3. ***Multi-class Logistic Regression:***
   Finally, $\boldsymbol{V}(s,t)$ is transformed into the state output probabilities, $P(s|\boldsymbol{o}_t)$, by means of regression. In this work, logistic regression is used, where the transformation is given by the following softmax function:

$$P(s|\boldsymbol{o}_t) = \frac{\exp\left(\boldsymbol{w}_{c(s)}^T \cdot \boldsymbol{V}(s,t)\right)}{\sum_{s' \in S} \exp\left(\boldsymbol{w}_{c(s')}^T \cdot \boldsymbol{V}(s',t)\right)} \qquad (1)$$

   where S is the set of the possible states and $c(s) \in C$ is the triphone state cluster for state s. The state clusters in C are obtained from the conventional phonetic decision trees [1]. The regression weights, $\boldsymbol{w}_c$, are defined for each state cluster, c, to reduce the number of free parameters. Although the regression parameters are shared by the states within a state cluster, the $\boldsymbol{V}(s,t)$ term will result in a different state output probability since $\boldsymbol{D}_s$ is unique for each state. Since the denominator of equation 1 is independent of s, this term is just a constant bias which can be ignored during decoding. Therefore, the log probability of each state can be easily computed as a simple dot product:

$$\log P(s|\boldsymbol{o}_t) \propto \boldsymbol{w}_{c(s)}^T \cdot \boldsymbol{V}(s,t) \qquad (2)$$

   The dot product can be computed efficiently since the dimension of the vectors is low.

The CD regression itself can be viewed as a 2-layer NN with sparse connections (see Figure 2). The input units are given by the canonical state vector, $\bar{\boldsymbol{b}}_t$, and the output regression targets are all the possible CD states. The descriptor, $\boldsymbol{D}_s$, constrains that each CD state output is connected to only a very small number of hidden nodes. Therefore, despite the very high dimension of $\bar{\boldsymbol{b}}_t$ and $P(s|\boldsymbol{o}_t)$, the computation is actually quite cheap and can be computed dynamically on demand during decoding.

NNs are widely used in the NN/HMM systems as a "merger" [20] to provide the posteriors for decoding. However, it is important to note that the proposed regression-based CD DNN differs substantially from the merger configuration:

- A merger is usually a fully connected NN that attempts to retain as much information as possible from the high dimensional input features, whereas the proposed method uses a sparsely connected network to combine multiple detectors.
- The outputs of a merger are usually the state clusters, similar to the standard (D)NN/HMM hybrid systems. Therefore, they do not address the *clustering* problem explicitly.
- The regression NN requires two labels for training given a frame, the triphone state label and the state cluster label. The training of the "merger" can only use the state cluster label since its output posteriors are used for decoding, where the triphone state identity is *not* available.

The proposed work is also different from the detector-based automatic speech recognition approach [14, 15, 16], where NN [14] or DNN [15] with binary outputs are used to detect the articulator attributes in speech and a merger is used to combine various attributes to predict the state output posterior. The detector-based methods do not address the state clustering problem and cannot be easily scaled up to handle context-rich attributes. These methods focus on using the articulator attributes as an intermediate representation of the phonemes.

Place of articulation		Production manner		Voicedness		Miscellaneous	
Front Vowel	iy ih eh ae aw ey y	High Vowel	ih iy uh uw	Voiced	iy ih eh ey ae aa aw ay ah ao oy ow uh uw er b d dh g jh l m n ng r v w y z zh	Short Vowel	eh ih uh ae ah y oy
		Mid Vowel	ah eh ey ow er			Long Vowel	iy uw aa
Central Vowel	ah er hh					Diphthong	ey aw ow ao
Back Vowel	aa ao uh uw ay ow oy	Low Vowel	aa ae aw ay oy ao			ay	ay
Coronal	d l n s t z r th dh	Fricative	jh ch s sh z f zh th v dh hh			Retroflex	er r
						Affricate	ch jh
Palatal	sh zh jh ch	Nasal	m n ng			Alveolar	s z t d n l
Labial	b f m p v w	Stop Cons	b p t d k g	Unvoiced	p f th t s sh ch k hh	Continuent	sh th dh hh m f ng v w zh
Velar	g k ng	Approximant	w y l r			NonContinuent	p b g k
Silence	sil	Silence	sil	Silence	sil	Silence	sil

Fig. 3. *Broad phone classes based on place of articulatory (A), production manner (M), voicedness (V) and miscellaneous (O)*

3. CANONICAL STATE REPRESENTATION

It is important to design a good canonical state representation so that the context information can be modelled efficiently. As previously mentioned in Section 2, a canonical state vector is represented by a series of front-end detectors. In this work, each detector is responsible to predict the posterior probabilities of biphone clusters. Instead of automatically generating the biphone clusters using decision tree clustering in a data-driven manner [1], we used the "board phone" classes to cluster biphone contexts. Broad phones are subphonetic articulatory features that can be used to describe phonemes from multiple perspectives. In this paper, we introduce four categories of articulatory features based on the place of articulation $A(\cdot)$, production manner $M(\cdot)$, voicedness $V(\cdot)$ and miscellaneous $O(\cdot)$ as shown in Figure 3. The miscellaneous category is designed to discriminate phones that cannot be distinguished by the other three categories. Each phone appears *only once* in one group. By considering both the left and right biphones, we have a total of eight biphone clusters. Therefore, eight DNNs are trained to predict the posterior probabilities of these biphone clusters.

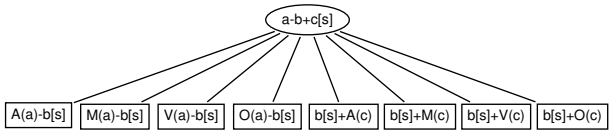

Fig. 4. *Illustration of broad phone contexts for "a-b+c[s]"*

A state descriptor, $\boldsymbol{D}_s$, is hence an 8-dimensional vector indicating the cluster that the state s belongs to in each of the 8 biphone clusters. The design principle of the broad phone classes is to assign each triphone state to a *unique* descriptor that is composed of simpler biphones clusters, which are easier to train and predict. This also addresses the data sparsity problem of predicting a single set of triphone state clusters directly in the standard DNN configuration. Figure 4 illustrates the broad phone contexts of a triphone state "a-b+c[s]". For example, the biphone clusters for state "sh-iy+n[2]" are given by "{palatal-iy[2], fricative-iy[2], unvoiced-iy[2], continuent-iy[2], iy[2]+coronal, iy[2]+nasal, iy[2]+voiced, iy[2]+aleveolar}" since phone /sh/ has the property of palatal, fricative, unvoiced and continuent; /n/ has the properties of coronal, nasal, voiced and aleveolar. Each element of the descriptor, $\mathbf{D}_{\text{sh-iy+n[2]}}$, holds the cluster index of the corresponding biphone cluster.

4. REGRESSION PARAMETER ESTIMATION

A straightforward way of computing $P(s|\boldsymbol{o}_t)$ is to simply add the log posterior probabilities of the corresponding biphone clusters using uniform weights. This corresponds to setting $\boldsymbol{w}_c$ to be $1/N$ and no additional learning of the regression parameters is needed. In fact, the interpolation with uniform weights already gives promising improvements as reported in Section 5. Nevertheless, it is possible to obtain further improvement by learning the regression weights from the training data. In the following, two methods to estimating the regression parameters will be described.

4.1. Logistic Regression

The regression model proposed in the previous section can be estimated by minimising the cross-entropy between the target state label vector, $y_t(s)$ and the state output probabilities predicted by the model:

$$\mathcal{F}_{\text{XENT}} = -\sum_{t}^{T}\sum_{s\in S} y_t(s)\log P(s|\boldsymbol{o}_t) = -\sum_{t}^{T}\log P(s_t|\boldsymbol{o}_t)$$

where in the case of hard target labels $y_t(s) = 1$ if $s = s_t$ and $y_t(s) = 0$ otherwise. s_t is the correct state label at time t. Substituting equation 1 into the above objective function yields:

$$\mathcal{F}_{\text{XENT}} = -\sum_{t}^{T}\left\{\boldsymbol{w}_{c(s)}^{T}\cdot\boldsymbol{V}(s,t) - \log\boldsymbol{Q}_{s_t}\right\} \tag{3}$$

where

$$\boldsymbol{Q}_{s_t} = \sum_{s'\in S}\exp\left(\boldsymbol{w}_{c(s')}^{T}\cdot\boldsymbol{V}(s',t)\right) \tag{4}$$

S denotes a set of all the triphone states. It is not feasible to directly optimise $\mathcal{F}_{\text{XENT}}$ in equation 3 because it involves the summation over all the states, many of which do not have enough training data. Furthermore, it will be computationally intractable to compute the summation over all the states during training. To circumvent this problem, instead of computing $\boldsymbol{V}(s,t)$ for all the states, we compute only one state, s_c, for each state cluster c. The rest of the states in that cluster will use the CD state vector of s_c when computing the objective function. s_c can be viewed as a representative state for cluster c. Therefore, the new objective function, $\mathcal{F}'_{\text{XENT}}$ can be obtained by replacing $\boldsymbol{Q}_{\boldsymbol{s_t}}$ in equation 4 with $\boldsymbol{Q}'_{\boldsymbol{s_t}}$:

$$\boldsymbol{Q}'_{s_t} = \sum_{s'\in S}\exp\left(\boldsymbol{w}_{c(s')}^{T}\cdot\boldsymbol{V}(s_{c(s')},t)\right) \tag{5}$$

$$= \sum_{c\in C} N_c\exp\left(\boldsymbol{w}_{c}^{T}\cdot\boldsymbol{V}(s_c,t)\right) \tag{6}$$

where $s_{c(s')}$ is the representative state of the cluster that the state s' belongs to. C is the set of state clusters and N_c is the number of states in cluster c. We further constrain that $\boldsymbol{V}(s_t, t)$ for the reference state s_t is computed directly and will not use the representative state approximation. Note that $\boldsymbol{Q}'_{s_t}$ can be computed more efficiently since the summation is now over all the state clusters.

Next, it is necessary to make sure that $\mathcal{F}'_{\text{XENT}} \geq \mathcal{F}_{\text{XENT}}$ so that minimising $\mathcal{F}'_{\text{XENT}}$ will result in a decrease in $\mathcal{F}_{\text{XENT}}$. This can be achieved by finding the cluster state representatives, s_c, such that $\boldsymbol{Q}'_{s_t} \geq \boldsymbol{Q}_{s_t}$. One simple solution is to constrain the regression weights, $\boldsymbol{w}_c$ to be positive and set a different cluster representative for each element of $\boldsymbol{V}(s', t)$ so that

$$s_c[i] = \arg\max_{s' \in c(s')} \boldsymbol{V}(s', t)[i], \quad \text{where } i \in [1..N]$$

Therefore, for all i

$$\boldsymbol{V}(s_c, t)[i] \geq \boldsymbol{V}(s', t)[i] \quad \Rightarrow \quad \boldsymbol{Q}'_{s_t} \geq \boldsymbol{Q}_{s_t} \tag{7}$$

since $\boldsymbol{w}_c$ and N_c are nonnegative. However, computing the representative state, s_c, in this way is still computational expensive since the algorithm still needs to go through all the states in the *max* operation for each frame t.

To mitigate this issue, we further constrain the representative states to be static (frame independent) so that it can be obtained once and reused in subsequent optimisation iterations. We propose to choose the state with the largest number of training frames to represent the state clusters. The rational is that the shared weights corresponding to the output state cluster $c(s')$ are trained using the frames from all its triphone state members s'. The triphone state with the largest number of training frames "contributes" the most to the shared weights $\boldsymbol{w}_{c(s')}$. In other words, the weights are trained so that the acoustic property of the cluster is closest to the one with the largest number of training frames. If we further assume that all the state clusters have the same number of states, N_c can be omitted in equation 6 and the objective function simplifies to a standard multi-class logistic regression where the target classes are given by the cluster representatives:

$$\mathcal{F}'_{\text{XENT}} = -\sum_{t}^{T} \left\{ \mathbf{w}_{s_t}^T \cdot \mathbf{V}(s_t, t) + \log \sum_{c \in C} \exp\left(\mathbf{w}_c^T \cdot \boldsymbol{V}(s_c, t) \right) \right\}$$

Therefore, the problem of optimising the regression model for a full set of CD states has been approximated with one that optimises for the state clusters. Although this approximation does not guarantee to minimise the original cross-entropy objective function, it has been found empirically to work well and yield promising improvement (see Section 5).

4.2. Nonparametric Frame-varying Regression

Alternatively, we propose a more flexible solution to compute the weights in a nonparametric fashion without the need for prior training. This method does not require the weights to be clustered and computes them on the fly based on the broad phone DNN posterior distributions. As a result, the regression weights become time dependent. Since each regression weight corresponds to a front-end detector, we propose setting higher weights for the detectors with a sharper posterior probability distribution. The rational is that a sharper posterior probability distribution indicates that the detector produces a more confident prediction and hence should be given a stronger emphasis. The nonparametric time-varying weights are computed as follows:

$$w_t[i] = \frac{\mathcal{KLD}_{ti}}{\sum_{k=1}^{N} \mathcal{KLD}_{tk}} \tag{8}$$

where $\mathcal{KLD}_{ti}$, the sharpness of the distribution, is measured in terms of the KL divergence of the ith detector's posterior distribution from the uniform distribution at time t:

$$\mathcal{KLD}_{ti} = \sum_{j=1}^{\mathcal{N}_i} P_{ij} \log \frac{P_{ij}}{\frac{1}{\mathcal{N}_i}} = \sum_{j=1}^{\mathcal{N}_i} P_{ij} \log P_{ij} + \log \mathcal{N}_i \tag{9}$$

where $\mathcal{N}_i$ is the number of posteriors of the ith detector and $P_{ij} = \boldsymbol{b}_{t,i}[j]$ is the jth posterior probability of the ith detector.

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

We evaluate the proposed CD modelling schemes for DNNs on a large and challenging broadcasting news transcription task using the Topic Detection and Tracking - Phase 3 (TDT3) corpus. [1] The English portion consists of approximately 475 hours of speech. Note the TDT 3 corpus is not carefully transcribed. The closed-captions of the corpus only have the time boundary information for the changes of topics or stories. Therefore, the corpus has to be pre-processed before use. The pre-processing includes: 1) removing non-speeches 2) normalising the closed captions and filtering of stories 3) segmenting audios into shorter utterances. After the preparation steps, 100 hours of speech data is left for acoustic model training.

The phone set contains 40 phones including silence. Each phone HMM is modelled with 5 states including 3 emitting states. The features are the standard 39-dimensional PLPs consisting of 13 static coefficients and the first and second derivatives. Each triphone state in the baseline GMM/HMM is modelled with 20 components. The testing set is the F0 portion of the Hub4-97 evaluation set. The language model is obtained from an interpolation of 2 language models trained with HTK [2] using the Gigaword English corpus and the TDT3 transcriptions respectively with a 58K vocabulary list. The perplexities on the Hub4-97 transcription are 295 and 201 for bigram and trigram respectively. For the DNN training, 10 hours of speech is separated as the cross-validation set. In addition, 4 hours of speech is separated from the cross validation set as the development set to tune the training and decoding parameters. The word error rate is obtained from a bigram full decoding and a trigram lattice rescoring.

5.2. Baseline Systems

The best performance of the baseline GMM/HMM is achieved with roughly 4500 clusters with the bigram WER of 28.3% and trigram WER of 23.1%. 4 iterations of MMI training [21] is performed thereafter, yielding a bigram WER of 25.8% and trigram WER of 20.9%.

DNN training is performed using the TNet [3] with an Nvidia Tesla M2090 GPU with 4G memory and 512 cores. Up to 5 hidden layers with 2048 hidden units are trained. Four 5-layer CD-DNNs with a varying target number are then fine-tuned with the pre-trained weights. The training labels are obtained from the forced alignments using the corresponding baseline GMM/HMMs. A CI DNN is also trained with 120 monophone states as training targets. The decoding of the hybrid DNN/HMM system is implemented using Kaldi [22].

[1] http://projects.ldc.upenn.edu/TDT3/

[2] http://htk.eng.cam.ac.uk/

[3] http://speech.fit.vutbr.cz/software/neural-network-trainer-tnet

The WER performance of the CI-DNN and CD-DNNs on the test set is compared in Table 1.

Table 1. WER performance of the CI-DNN and CD-DNNs

#clusters	120 (CI)	1601	2303	3052	4451
Bigram	20.0	18.9	18.4	18.5	18.5
Trigram	16.0	15.0	14.8	14.9	15.0

We can see that all the CD-DNNs outperform the CI-DNN, which justifies the importance of incorporating contexts for DNN training. Even the CI-DNN has a significantly better performance than the best baseline state clustered triphone GMM/HMM system with MMI training. However, different from the GMM/HMM systems, the best CD-DNN has 2303 state clusters rather than 4451. This indicates that with a larger number of state clusters, although the context resolution is finer, the back-propagation training is also more prone to trap into a poor local optimum even for DNNs.

As an initial attempt to address the clustering problem, two more DNNs were trained to predict the left and right contexts and factorize the triphone state to three DNN posteriors:

$$p(\text{a-b+c[s]}|\mathbf{o}_t) = p(\text{b[s]}|\mathbf{o}_t)p(\text{a}|\mathbf{o}_t)p(\text{c}|\mathbf{o}_t) \quad (10)$$

The best WER performance for this Bayesian factorisation is 18.2% for bigram and 14.6% for trigram. Significant test using the trigram results is performed using SCTK [4]. The trigram performance is marginally better than the best CD-DNN with 2303 state clusters.

5.3. CD DNN with the NN Logistic Regression Model

8 broad phone DNNs with 5 hidden layers are trained using the pretrained weights to predict the left and right biphones with broad phone contexts for the 4 grouping. We denote the 8 biphone context DNNs as "B(L)-S" and "S+B(R)", where B is a broad phone grouping mapping a phone L/R to its broad phone class, S is the monophone state. The output targets of the broad phone DNNs are the combinations of central monophone states and left or right broad phone contexts. The output layer size for the 8 broad phone DNNs is tabulated in Table 2:

Table 2. Output dimensions of the broad phone DNNs

A(L)-S	M(L)-S	V(L)-S	O(L)-S
S+A(R)	S+M(R)	S+V(R)	S+O(R)
939	939	353	1173

The WER of the hybrid DNN/HMM system using the biphone context DNNs is given in Figure 5. None of the 8 DNNs outperforms the best CD-DNN with 2303 clusters. This is expected since these DNNs only model one side of the contexts while for the decision tree clusters, the left and right contexts are jointly considered during clustering. Furthermore, the number of state clusters is significantly smaller than the best CD-DNN configuration.

For the logistic regression models, the input of the 2-layer regression NN is the concatenation of all the log posteriors from the 8 broad phone DNNs. Therefore, the input layer has totally $(939 + 939 + 354 + 1173) \times 2 = 6810$ units. The best GMM/HMM system has 4451 state clusters, whereas the standard CD DNN with 2303 state clusters has the best performance. Therefore, we trained two NNs with these two state cluster configurations. In addition, another 2-layer NN is also trained with the monophone states as the targets.

[4] http://www.nist.gov/speech/tools

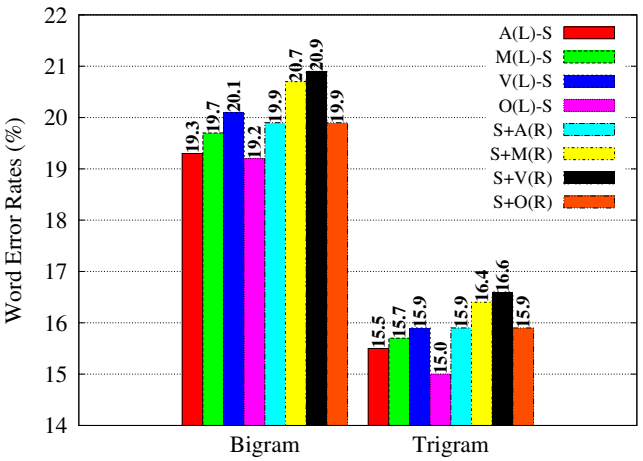

Fig. 5. *Broad phone context DNNs WER performance comparison*

The fine-tuning cross-validation (CV) accuracies for CD DNNs are 58.3% for 2303 clusters and 55.6% for 4451 clusters. The CV accuracies for the 2-layer regression NN are significantly better than their CD DNN counterparts, with 62.7% for 2303 clusters and 59.8% for 4451 clusters. This is expected since the training of the regression NN takes two types of information: the triphone state and the state cluster. Therefore, the cluster is much more easier to predict due to the additional triphone state information. The purpose of the 2-layer NN is *not* to provide the posteriors of the state clusters. Instead, the regression weights are used to combine the 8 broad phone DNN according to equation 2 so that each triphone state has a unique acoustic score for decoding.

The WER of the NN regression model is given in Figure 6. Even with only 120 CI regression targets, the NN regression with the CI states (NN-CI-Regre) has a bigram WER of 15.7% and trigram WER of 12.7%. The performance is significantly better than the frame-varying weights with a p-value smaller than 0.001. This clearly shows the advantage of the proposed NN regression scheme to discriminate all the triphone states. The NN regression model with 2303 clusters (NN-2303-Regre) has a bigram WER of 15.6% and trigram WER of 12.3%. This performance is significantly better than NN-CI-Regre. This indicates the importance of relaxing the context overlaps for the regression targets. The NN regression model with 4451 clusters (NN-4451-Regre) has a slightly better performance than NN-2303-Regre with bigram WER of 15.7% and trigram WER of 12.1%.

The regression NN differs substantially from the "merger" NN since it uses the triphone state information to dynamically change the weight connections during training. Since the triphone state is not available during decoding, it cannot be used to train the merger. Therefore, the outputs of the regression NN are *irrelevant* to and cannot be used for decoding. To verify this, the outputs of the regression NN are used for decoding as in the merger, the best WER with 4451 clusters is 20.3% for bigram and 15.8% for trigram. The performance is even significantly worse than the standard CD DNN.

5.4. Summary

The WER of the best configurations of the CI-DNN, CD-DNN, naive Bayesian factorisation, and the broad phone based CD DNN are compared in Figure 6. There are three broad phone DNN based CD DNNs, including the interpolation with uniform weights and the frame-varying weights, the logistic regression based CD DNN.

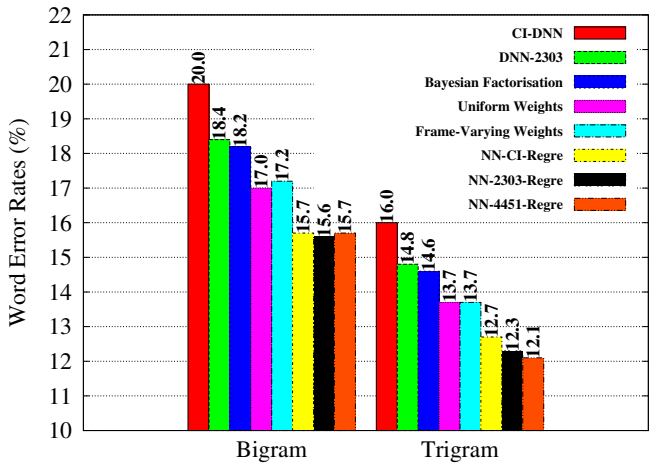

Fig. 6. *WER comparison of CD DNN modelling schemes*

From Figure 6, we can see that all the broad phone based CD DNNs outperform the best standard CD DNN with 2303 state clusters (DNN-2303). As an initial attempt to address the clustering problem, Bayesian already performs marginally better than DNN-2303. The canonical state modelling based NN regression has the best performance among the 3 broad phone based CD DNNs. All the regression-based CD DNNs outperform DNN-2303 significantly at 0.05 significant level.

The second CD-DNN (DNN-2303) trained with state clusters is how the current DNN literature handles the CD modelling problem. Compared to the CI-DNN, it does provide a significant performance gain. However, it does not consider the clustering problem and does not match the performance of the regression-based CD DNNs. Compared with DNN-2303, the best NN regression model NN-4451-Regre offers a 2.7% absolute WER reduction. This clearly shows the importance of addressing the clustering problem. For the regression-based CD DNNs, the broad phone DNNs are designed to tackle the data sparsity problem and define the canonical state space. With the NN regression model, the triphone states can be better modelled since the objective function is to maximise the context resolution among all the triphone states. Therefore, with the NN regression weights, each triphone state can be discriminated to each other during decoding with a unique acoustic score computed from the canonical states modelled by the broad phone DNNs under the canonical state modelling framework.

6. CONCLUSIONS

In this paper, a novel context-dependent (CD) modelling framework for Deep Neural Network (DNN) is proposed to address both the data sparsity problem and the clustering problem. The regression-based CD DNN is formulated as an instance of the canonical state modelling (CSM) technique. The triphone states are clustered into multiple sets of shorter biphones using broad phone contexts to address the data sparsity issue. The concatenated log posteriors of the broad phone clusters form the the canonical state vectors. A logistic regression model is trained to transform the canonical state vectors into triphone state probabilities to mitigate the clustering problem. The proposed regression-based CD DNN is evaluated on a broadcast news transcription task. Regression-based CD DNNs consistently outperform the baseline standard CD DNNs. In addition, the best configuration of the proposed CD DNN with 4451 regression targets provides a significant performance gain over the best standard CD DNN system with a 2.7% absolute WER reduction.

7. ACKNOWLEDGMENT

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

8. REFERENCES

[1] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *HLT*, 1994, pp. 307–312.

[2] N. Morgan and H. Bourlard, "Neural networks for statistical recognition of continuous speech," *Proc. IEEE*, vol. 83, pp. 742–772, 1995.

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

[4] G. Wang and K. C. Sim, "Comparison of smoothing techniques for robust context dependent acoustic modelling in hybrid NN/HMM systems," in *Interspeech*, 2011, pp. 457–460.

[5] ——, "Sequential classification criteria for NNs in automatic speech recognition," in *Interspeech*, 2011, pp. 441–444.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

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

[8] G. E. Dahl, D. Y. L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. on ASLP*, vol. 20, pp. 30–42, 2012.

[9] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*, 2011, pp. 24–29.

[10] T. N. Sainath, B. Kingsbury, and etc, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011, pp. 30–35.

[11] G. Hinton, L. Deng, D. Yu, and etc, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[12] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 14–22, 2012.

[13] M. J. F. Gales and K. Yu, "Canonical state models for automatic speech recognition," in *Interspeech*, 2010, pp. 58–61.

[14] C.-H. Lee, M. Clements, S. Dusan, E. Fosler-Lussier, and etc, "An Overview on Automatic Speech Attribute Transcription (ASAT)," in *Proc. InterSpeech*, 2007.

[15] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomput.*, vol. 106, pp. 148–157, Apr. 2013.

[16] L. Deng and D. Sun, "A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features," *J. Acou. Soc. Am.*, vol. 95, no. 5, pp. 2702–2719.

[17] D. Povey and L. B. et al., "The subspace gaussian mixture model-a structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, pp. 404–439, 2011.

[18] M. J. F. Gales, "The generation and use of regression class trees for MLLR adaptation," Cambridge University Engineering Department, Tech. Rep., 1996.

[19] K. Yu, H. Zen, F. Mairesse, and S. Young, "Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis," *Speech Commun.*, vol. 53, no. 6, pp. 914–923, 2011.

[20] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Brno University of Technology, Faculty of Information Technology, 2008.

[21] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young, "MMIE training of large vocabulary recognition systems," *Speech Comm.*, vol. 22, no. 4, pp. 303 – 314, 1997.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, and etc, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.