CONTEXT-DEPENDENT CLASSES IN A HYBRID RECURRENT NETWORK-HMM SPEECH RECOGNITION SYSTEM

D. J. Kershaw, M. M. Hochberg & A. J. Robinson

CUED/F-INFENG/TR217

July 1995

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

CONTEXT-DEPENDENT CLASSES IN A
HYBRID RECURRENT NETWORK-HMM
SPEECH RECOGNITION SYSTEM

D. J. Kershaw, M. M. Hochberg & A. J. Robinson

CUED/F-INFENG/TR217

July 1995

Abstract

A modular method for incorporating context-dependent phone classes in the CUED connectionist-HMM hybrid speech recognition system is introduced. The current CUED connectionist-HMM hybrid system performs well on large vocabulary speech recognition tasks. Although the recurrent framework does model acoustic context internally (mainly in the hidden state vector), the targets are currently context independent. It is proposed that by including phonetic-context dependent targets to the recurrent network, improved modelling would be possible, as is seen in equivalent monophone and triphone HMM systems.

This report discusses the methods necessary to introduce context-dependent outputs into the hybrid system. It focusses on two main issues: Which context classes should be modelled and which would be best for the recurrent framework, and given a set of context classes which mechanism should be employed to model them. A decision-tree based approach was used to cluster the different context classes of a phone. The final training strategy involved a modular solution, whereby single-layer networks were trained on the state-vector to discriminate between the different context classes, given the phone class.

Some initial experiments show an average reduction of around 16% in word error rate on some ARPA Wall Street Journal tasks. The new context-dependent system still has far fewer parameters than any equivalent HMM system, and due to improved modelling decoding speed is over twice as fast as the context-independent system.

Contents

1 Introduction

The ABBOT hybrid connectionist-HMM system performed competitively with many conventional hidden Markov models (HMM) systems at the recent ARPA evaluations. This hybrid framework is attractive because it performs fast sentence decoding and is compact, having far fewer parameters than conventional HMM systems. A simple overview of the ABBOT system follows in Section 1.1.

State-of-the-art HMM systems employ some form of modelling of phones-in-context to account for the acoustic variability of a phone given its context. This modelling of phones in their particular phonetic contexts vastly improves their performance over equivalent context-independent HMM systems (see Section 1.2). Although the recurrent neural network (RNN) does model acoustic context internally, it does not model phonetic context; the target outputs are only context independent.

It is proposed that the ABBOT connectionist-HMM hybrid system could have improved phonetic modelling by providing context-dependent output targets. This report describes :-

- a context-class selection strategy,
- training issues involved in building such a context-dependent system,
- a full context-dependent modular architecture description,
- evaluation on several ARPA Wall Street Journal test sets.

1.1 A Simple Overview of the ABBOT System.

Connectionist systems have been successfully used in the area of speech recognition. The time-delay neural network (TDNN) has been reported to perform well on phoneme recognition tasks [21], while the multi-layer perceptron (MLP) and the recurrent neural network have demonstrated their practical use in the area of large vocabulary speech recognition [18]. The basic framework of the ABBOT system is similar to the one described in [2], except that a recurrent network is used for the connectionist component. A more detailed description of the recurrent network for phone probability estimation is given in [17] [19].

1.1.1 The Recurrent Neural Network

A recurrent network in Figure 1 is used as the acoustic model within the HMM framework. At each 16ms time frame, the input acoustic vector is mapped to an output vector $\mathbf{y}(t)$. Two forms of spectral input representation that have been found to be effective are :-

- MEL+ — a 20 channel mel-scaled filter bank with voicing, pitch and power parameters
- PLP — 12th order cepstral coefficients derived from perceptual linear prediction plus energy.

The output vector represents an estimate of the posterior probability of each of the phone classes

$$y_i(t) \simeq \Pr(q_i(t)|\mathbf{u}_1^{t+4}) \tag{1}$$

where $q_i(t)$ is phone class i at time t, and $\mathbf{u}_1^t = \{\mathbf{u}(1), \ldots, \mathbf{u}(t)\}$ is the input from time 1 to t. Left (past) acoustic context is modelled internally in a 256 dimensional state vector $\mathbf{x}(t)$, which can be envisaged as "storing" the information that has been presented at the input. Right (future) acoustic context is given by delaying the posterior probability estimation until four frames of input has been seen by the network. The network is trained using a modified version of error back-propagation through time [17].

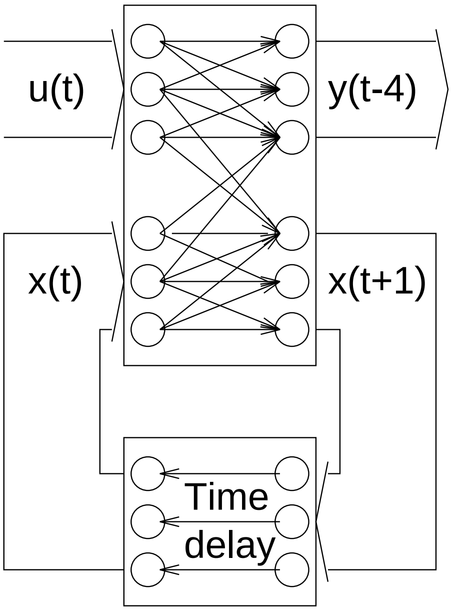

Figure 1: The recurrent neural network for phone probability estimation

1.1.2 Hybrid Utterance Decoding

Decoding with the hybrid connectionist-HMM approach is equivalent to conventional HMM decoding, with the difference being that the RNN provides the modelling of the observations. Like typical HMM systems, there exists a mapping between a state sequence Q on a discrete, first-order Markov chain and the word sequence. This allows expression of the recognition process as finding the maximum a posteriori (MAP) state sequence of length T,

$$\hat{Q} \quad = \quad \underset{Q}{\operatorname{argmax}} \prod_{t=1}^{T} \Pr(q_i(t)|q_i(t-1))\mathrm{p}(\mathbf{u}(t)|q_i(t)). \tag{2}$$

The decoding criterion specified above requires the computation of the likelihood of the acoustic data given a phone (state) sequence, $\mathrm{p}(\mathbf{u}(t)|q_i(t))$. Using Bayes' theorem,

$$\mathrm{p}(\mathbf{u}(t)|q_i(t)) = \frac{\Pr(q_i(t)|\mathbf{u}(t))\mathrm{p}(\mathbf{u}(t))}{\Pr(q_i)} \tag{3}$$

where $\mathrm{p}(\mathbf{u}(t))$ is the same for all phones, and hence drops out of the decoding process. Hence the network outputs are mapped to scaled likelihoods by,

$$\mathrm{p}(\mathbf{u}(t)|q_i(t)) \simeq \frac{y_i(t)}{\Pr(q_i)} \tag{4}$$

where $\Pr(q_i)$ is estimated from the training data. Rewriting equation 2 in terms of the network outputs therefore yields,

$$\hat{Q} \quad = \quad \underset{Q}{\operatorname{argmax}} \prod_{t=1}^{T} \Pr(q_i(t)|q_i(t-1))\frac{y_i(t)}{\Pr(q_i)}. \tag{5}$$

The decoding can be seen as a Markov process, determined in a hierarchical fashion, such that the language model (trigram, bigram or word pair) is a Markov process on the words, and the words are a Markov process on the phones (strings of phone models form words in the lexicon). Decoding uses the NOWAY decoder to compute the utterance model that was most likely to have generated the observed speech signal. Details of this decoder, and its operation can be found in [8]. Figure 2 shows the full system, with all its components, from the input speech waveform, to the decoded utterance. This is a simple representation of current systems.

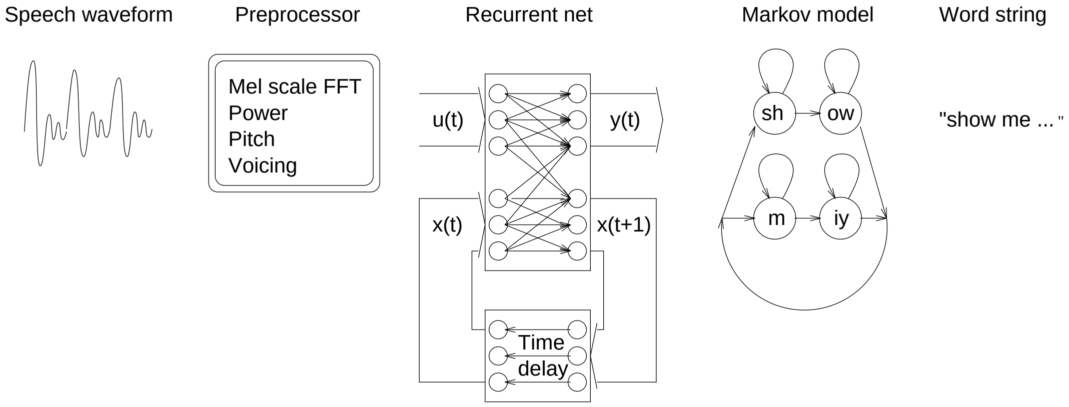

Figure 2: The full ABBOT Hybrid System

1.2 Using Phonetic Context To Reduce Word Error Rates

It is well established that particular phones vary acoustically when they occur in different phonetic contexts. For example a vowel may become nasalized when following a nasal sound. Context-dependent modelling of phones is powerful because it models the most important co-articulatory effects, and is therefore more sensitive than context-independent modelling. This is primarily achieved in HMM technology by modelling phonetic context. The short-term contextual influence of co-articulation is handled by creating a model for all sufficiently differing phonetic contexts with enough acoustic evidence. It produces models with sharper probability functions. This approach vastly improves HMM recognition accuracy over equivalent context-independent systems, and can be seen in HMM systems [15] and [22], and is summarised in Table 1.

System	CI % Word Error	CD % Word Error	% Reduction in WER
SPHINX	15.6	6.7	57.1
HTK	11.3	5.7	49.6

Table 1: Word Error Rates on DARPA Resource Management Feb89 Task, for the SPHINX System and the HTK system. Both systems use a word-pair grammar. Results for the HTK system are for the two-mixture monophones, and the two-mixture tied-state triphones, and are not the best systems available, but are included for CD and CI comparison only.

A comparison of other HMM systems (all with phonetic context models, and some with cross-word phonetic models) with the ABBOT system at the 1994 ARPA North American Business News evaluation [13] can be seen in Table 2.

The ABBOT system, although competitive, fairs less well than many of the context-dependent HMM systems. Given the improvement seen in HMMs in moving from context-independent to

Site	% Word Error
CU-HTK	10.5
IBM	11.1
BBN	11.9
Limsi	12.1
SRI	12.2
AT&T	13.0
BU-BBN	13.0
NYU-BBN	13.0
Dragon	13.2
Philips	13.4
CMU	13.7
ABBOT	**14.1**
MIT-LL	19.0
CRIM	20.2
KU	22.8

Table 2: Adjudicated Word Error Rates on the ARPA November 1994 H1 C1 (1994) standard 20k trigram Language Model Evaluation.

context-dependent systems, and the fact that the ABBOT system is context-independent (although it models acoustic context internally)[1], it was postulated that a reduction in word error rate was possible by modelling phonetic context at the output targets of the RNN.

The rest of this report investigates the options available, the methodologies used and preliminary results of modelling phonetic context in a recurrent neural network framework.

[1] HMMs also model acoustic context by using delta and acceleration parameters.

2 Clustering Context Classes

One of the new problems faced by having a context-dependent system is to decide which context classes are to be included in the CD system. One choice available is a clustered triphone system, with a minimum number of frames per triphone. However this presents two problems: This method doesn't necessarily pick the best context classes for discrimination purposes and with large vocabulary speech recognition it is likely that context classes in testing do not occur in the training data. A proposed method for overcoming this problem is a decision-tree based approach to cluster the new context classes. This is extremely appealing because this guarantees a full coverage of all phones in any context, and the context classes are chosen using the acoustic evidence available. The tree clustering framework also allows for the building of a small number of context-dependent phones, keeping any new context-dependent connectionist system architecture compact. There are two stages to decision-trees: firstly how they are grown and secondly once grown, how they are used.

2.1 An Introduction to Growing Decision Trees

This approach to clustering contexts is quite common, and hence only a brief description will be given. The approach taken here is based on [3], [23] and [11]. Basically, each terminal in the tree has some data associated with it. A particular question is asked at a terminal node which would initially split the data into two child nodes (one for an answer of yes, and one for an answer of no). The *goodness of split scoring criterion* is then calculated for this preliminary split. All the questions are asked at all the terminals node making preliminary splits, and the terminal with the best *splitting score* **is** split, while the question asked to achieve this best *splitting score* is stored at the terminal that is split. The splitting continues until the best overall *goodness of split score* falls below some threshold. The *goodness of split scoring criterion* used is very much dependent on the representation of the data.

2.2 Decision Tree Based Approach to Clustering Context Classes

The following will outline the procedures taken to cluster class contexts. The acoustic data is in the form of a processed speech waveform (MEL+) which is assumed to be multivariate Gaussian, each channel having a mean of zero and a variance of one. Each channel of the preprocessed acoustic data is treated as independent. Since the data is assumed to be Gaussian, the node splitting criterion is based on log likelihood. The procedure is as follows:-

Rotate data – Diagonalise the acoustic data for each monophone class i, by rotating by the eigenvectors of the covariance matrix Σ_i.

$$U^T \Sigma_i U = \Lambda$$

By having "diagonalised data" computational requirements are signicantly reduced. No further rotations are performed.

Initialise trees – Initialise a tree for each monophone by putting all the rotated acoustic data in the root node.

Question set – The question set contains different phonetic groups ranging from Vowel, to Glide, to particular phones like /ax/ or /tcl/ etc. These can be asked in left or right context. For example, *is the left context Class-Fricative?*. Phonetic groups were chosen using [14] [16]. Examples of these can be seen in appendix A.

Grow trees – For each terminal in the tree calculate the diagonal covariance of the acoustic data associated with the node. Apply the tree growing algorithm as described in the previous section, using equation 6 as the *goodness of split scoring criterion.* Growing is stopped either when all split scores at each terminal for every possible question falls below the threshold, or when a certain number of terminals have been created. (This enables a further control on the total number of context classes created.)

Split Criterion – The goodness of split scoring criterion is based on the gain in log Likelihood due to the data being split, and simplifies to

$$\Delta L = n_p \log |\Sigma_p| - (n_{c1} \log |\Sigma_{c1}| + n_{c2} \log |\Sigma_{c2}|) \tag{6}$$

where Σ_p refers to the diagonal covariance of the rotated acoustic data at the parent node, Σ_{c1} and Σ_{c2} refers to the diagonal covariance of the rotated acoustic for the children, and n is the number of examples at a node.

2.3 Using the Context Trees

An example of a context-clustered tree is shown in Figure 3. Once built the terminal nodes of the tree are labelled. These labels represent the new context-classes of the monophone. These context-dependent labels now make up the context-dependent phone set. The trees were used to relabel the training data and the word lexicon. This was done by going to the phone-in-context's (i.e. the central phone) tree, looking at the immediate left and right context of the phone-in-context, and proceeding down the tree, answering the question at each node until a terminal node is reached. The central phone is then relabelled with the new label stored at the terminal node that the phone-in-context reached.

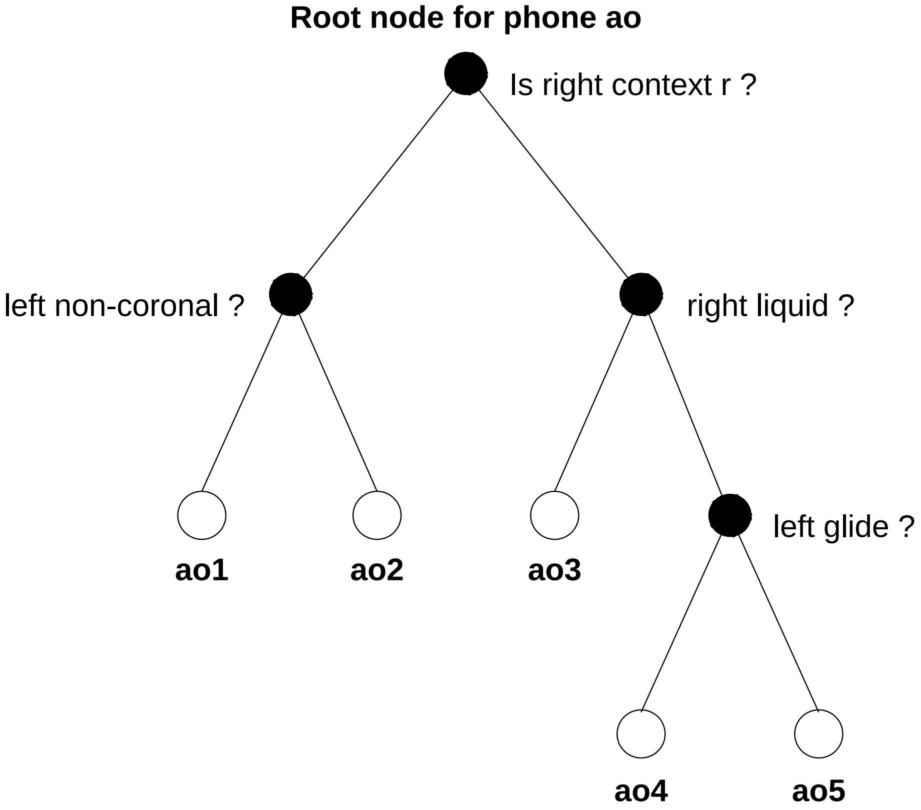

Figure 3: An example tree grown for phone class /ao/.

3 Previous Work on Context-Dependent Hybrid HMM-Connectionist Systems

This section briefly describes some of the previous work undertaken to incorporate phonetic context into a connectionist framework. Three systems are described:

- context-dependent modelling by MLP at SRI
- context-dependent modelling by MLP at ICSI
- the context-dependent segmental neural network (SNN) and hierarchical mixtures of experts,

3.1 Context-Dependent Modelling by MLP at SRI

One representation of the context-dependent phonetic modelling requires the computation of $p(\mathbf{U}_t|\mathbf{C}_t, \mathbf{Q}_t)$, the probability density of the acoustic data $\mathbf{U}$ given the phone class $\mathbf{Q}$ in the context class $\mathbf{C}$, at time t. This approach is adopted by [6], whereby context-specific networks are trained. This means that networks are trained to discriminate between phone classes given a context, i.e. estimating $p(\mathbf{Q}_t|\mathbf{C}_t, \mathbf{U}_t)$. This scheme makes use of a sharing of parameters for faster training times. Every context network performs a simpler classification task than in the CI case, because in a given context, the acoustic correlates of different phones have much less overlap in their class boundaries. However, this format also requires the training of a CI MLP to discriminate between different context class boundaries.

3.2 Context-Dependent Modelling by MLP at ICSI

Another approach to phonetic context-dependent modelling with MLPs was proposed in [1]. It was based on factoring the conditional probability of a phone-in-context given the data in terms of the phone given the data, and its context given the data and the phone. This implementation requires only one MLP estimator for each of left and right context are required. However, the left phonetic context MLP requires extra binary inputs of current state and right phonetic context, while the right phonetic context MLP requires extra binary inputs of current state. This is not a problem during the supervised training stage. However, during recognition a “contextual contribution” calculation is required.

3.3 Context-Dependent SNN and HME

The work presented in [24] augments the BYBLOS HMM System, using left-context Segmental Neural Networks (SNN). The HMM system is used to generate a segmentation for each N-best hypothesis, and the SNN used to generate a score for the hypothesis. The context-dependent training follows along similar lines as [6]. In training the left-context SNNs, the input is separated into 53 classes determined by the identity of the preceeding phoneme. Each left context SNN can then be trained in a similar manner to the CI SNN, but only on the subset of the training data that corresponds to the SNNs context. In the rescoring process, each segment score is obtained by combining the output of the CI SNN with the corresponding output of the SNN that models the left-context of the segment, in a fashion similar to the way context HMMs are combined with CD HMMs in the BYBLOS system. This method reflects a modular solution to building context-dependent connectionist systems. However, it does require the N-best scheme for the segmentation process.

Further work in [25] applied this context-dependent training process to one-level hierarchical mixtures of experts.

4 Context-Dependent Training and System Description

This section will describe the issues involved in training context-dependent classes in the recurrent neural network framework. A full system description of the modular approach taken, which is similar in many respects to [6], [4] and utilises the factorisation seen in [1]. The major issues of concern are fast training, simple architecture implementation, and a framework that allows use of the standard NOWAY decoder.

4.1 Architecture Issues

Context-dependent phonetic modelling requires the computation of $\mathrm{p}(\mathbf{U}_t|\mathbf{CD}_t)$, the probability density of the acoustic vector $\mathbf{U}$, given the context-dependent phone class $\mathbf{CD}$, at time t. Using Bayes' theorem as in equation 3

$$\mathrm{p}(\mathbf{U}_t|\mathbf{CD}_t) = \frac{\Pr(\mathbf{CD}_t|\mathbf{U}_t)\mathrm{p}(\mathbf{U}_t)}{\Pr(\mathbf{CD}_t)}, \tag{7}$$

where $\mathbf{CD}_t \in \{\mathbf{C}_t, \mathbf{Q}_t\}$, ie $\mathbf{CD}_t$ is phone $\mathbf{Q}_t$ in the context of $\mathbf{C}_t$. One possible implementation of equation 7 is to train a recurrent neural network with N context-dependent phone targets. However, this is impractical because of the training time involved. (On the ARPA WSJ SI84 training corpus, the CI RNN with only 54 target outputs takes approximately five days to train on the RAP (Ring Array Processor) [12] .)

The approach taken by this work is to augment the CI RNN, and discriminate between different context classes, given a phone class, $\Pr(\mathbf{C}_t|\mathbf{Q}_t, \mathbf{U}_t)$, in a similar vein to [1]. Using Bayes' theorem again,

$$\mathrm{p}(\mathbf{U}_t|\mathbf{C}_t, \mathbf{Q}_t) = \frac{\Pr(\mathbf{C}_t|\mathbf{U}_t, \mathbf{Q}_t)\mathrm{p}(\mathbf{U}_t|\mathbf{Q}_t)}{\Pr(\mathbf{C}_t|\mathbf{Q}_t)}. \tag{8}$$

Substituting for the context independent probability density function $\mathrm{p}(\mathbf{U}_t|\mathbf{Q}_t)$ and using the Bayes' expression in equation 3, the following expression is obtained for the likelihood of the acoustic data given the context-dependent phonetic class,

$$\mathrm{p}(\mathbf{U}_t|\mathbf{C}_t, \mathbf{Q}_t) = \frac{\Pr(\mathbf{C}_t|\mathbf{U}_t, \mathbf{Q}_t)\Pr(\mathbf{Q}_t|\mathbf{U}_t)}{\Pr(\mathbf{C}_t|\mathbf{Q}_t)\Pr(\mathbf{Q}_t)}\mathrm{p}(\mathbf{U}_t). \tag{9}$$

The term $\mathrm{p}(\mathbf{U}_t)$ is constant for all frames, so this drops out of the decoding process and is ignored for all further purposes. This format is extremely appealing. $\Pr(\mathbf{C}_t|\mathbf{Q}_t)$ and $\Pr(\mathbf{Q}_t)$ are estimated from the training data. The CI RNN estimates $\Pr(\mathbf{Q}_t|\mathbf{U}_t)$, so all that is needed is an estimate of $\Pr(\mathbf{C}_t|\mathbf{U}_t, \mathbf{Q}_t)$. A possibility is to train a set of *context experts* or *modules* for each monophone class, to augment the existing CI RNN System. The contexts for these *modules* are given by the tree clustering routine from the previous section.

4.2 Input Representation for Training Context

It is necessary to find a fitting representation of the feature space, simple network architecture and training algorithm, in order to extract the context classes with a good accuracy and a fast training time. Another concern is that the scheme chosen fits neatly into the recurrent framework that currently exists.

In [6] the context-specific networks are trained on a non-overlapping subset of the original training data. The "training data" used is the hidden layer (of the CI MLP) representation of the acoustic features. The underlying assumption here is that the hidden layer representation of the acoustic features is rich enough to allow accurate modelling of the class boundaries in the different contexts.

The same assumption is made in this report, about the hidden layer, or state vector, of the CI recurrent network. It is assumed that the CI RNN's state vector is well trained, and contains all the contextual information necessary to discriminate between different context classes of a phone. It is also hypothesised that the state vector splits the acoustic feature space into many different hyper-planes, and hence a simple network structure such as a single layer perceptron can be used to discriminate between the context classes in the context modules. This means that the context module training will be fast.

4.3 Training on the State Vector

From Section 4.1, an estimate of $\Pr(\mathbf{C}_t|\mathbf{U}_t, \mathbf{Q}_t)$ must be found. An estimate can be made of this by training a recurrent network to discriminate between contexts $c_j(t)$ for phone class $q_i(t)$, such that

$$y_{j|i}(t) \simeq \Pr(c_j(t)|\mathbf{u}_1^{t+4}, q_i(t)) \tag{10}$$

where $y_{j|i}(t)$ is an estimate of the posterior probability of context class j given phone class i. However, training recurrent neural networks in this format would be expensive and difficult. For a recurrent format, the network must contain no discontinuities in the frame-by-frame acoustic input vectors. This implies all recurrent networks for all the phone classes i must be "shown" all the data. Instead, the assumption is made that since $\mathbf{x} = f(\mathbf{u})$, and that the state vector contains all the important contextual information necessary to train the context experts, that

$$\underline{\mathbf{x}(t+4) \text{ is a good representation for } \mathbf{u}_1^{t+4}}$$

Hence a single-layer perceptron is trained on the state vectors corresponding to each monophone q_i, classifying into different phonetic context classes. Finally the likelihood estimates for the phonetic context class j for phone class i used by the NOWAY decoder is given by,

$$\begin{aligned} \mathrm{p}(\mathbf{u}(t)|c_j(t), q_i(t)) &\simeq \frac{\Pr(q_i(t)|\mathbf{u}_1^{t+4}) \Pr(c_j(t)|\mathbf{x}(t+4), q_i(t))}{\Pr(c_j(t)|q_i(t)) \Pr(q_i(t))} \\ &\simeq \frac{y_i(t) y_{j|i}(t)}{\Pr(c_j(t)|q_i(t)) \Pr(q_i(t))} \end{aligned} \tag{11}$$

4.4 Comparison With Previous Work

This approach is similar in many respects to the work highlighted in section 3. Differences and similarities are outlined below:

- this modular approach is applied in the recurrent neural network framework,
- this approach clusters the context classes and results in a system with far fewer parameters to estimate (allowing training on a workstation rather than a transputer like the RAP). Also the approach taken in [24] requires an N-Best HMM framework to generate the segmentations for the segmental neural network (SNN) classifier or for a modified hierarchical mixture of experts [25],
- while this approach does require a series of additional networks (or modules), no additional binary inputs (for the current phone state, and left or right context) are necessary. This means that extra processing during the recognition stage to estimate these inputs is required, as found in [1],
- the context modules calculate the context of a phone class given the data, rather than a phone class given the context and the data (conditioning on the context rather than the phone) as in [6], [4], which means that new networks must trained to estimate the broad left context given the data, and the broad right context given the data. This is not a requirement in the work presented here.

- the architecture has similarities with mixture of experts [10]. During training, rather than making a "soft" split of the data as in the mixture of experts case, the Viterbi segmentation selects one expert at every exemplar. This means only one expert is responsible for each example in the data. This assumes that the Viterbi segmentation is a good approximation to the segmentation/selection process. Hence, each expert is trained on a small subset of the training data, avoiding the computationally expensive requirement for each expert to "see" all the data. During decoding, the RNN is treated as a gating network, smoothing the predictions of the experts, in an analogous manner to a standard mixture of experts gating net.

4.5 Summary of Full Training Scheme

The following lists the procedures used to train a set of new phonetic context models to estimate $y_{j|i}(t)$.

1. Build decision trees for all monophones.
2. Use a Viterbi segmentation to align the monophone labels with the data, and the relabel the data using the decision trees.
3. Run the relabelled training data through a CI RNN, in a feed-forward fashion. Store the state vectors $\mathbf{x}(t+4)$ (along with their associated context class labels j) for each monophone class i at time t.
4. Train a single layer perceptron module for each monophone class i, using a gradient descent training strategy on the state vectors for monophone class i, to classify into the context classes j. The outputs of module i now represents an estimator for $y_{j|i}(t)$.

5 Phonetic Context-Dependent Decoding

Decoding for the phonetic context-dependent system proceeds in much the same way as the context-independent system. Utterance decoding was achieved using the NOWAY decoder, just as in the context independent case. However, a few minor differences are described below.

New Pronunciation Lexicon

The decoding used a new pronunciation lexicon. The new lexicon was built using decision trees to relabel the old context-independent lexicon with the new context-dependent labels. This was possible because the new system constructed was *word internal* context-dependent. (Note that the label word-boundary does exist, but as yet no specific "word-boundary" questions where asked during the tree construction stage.) This means that there is, as yet, no specific cross-word phonetic context modelling. Likewise a new HMM model set was built for each new context model. For more information about the HMM part of the hybrid system see [9].

Generating the New Phonetic Context Posterior Probabilities

The new frame-by-frame phonetic context posterior probabilities are required as input to the NOWAY decoder. These posterior probabilities were calculated from the numerator of equation 11. The calculation of the phonetic context posterior probabilities can be visualised as possessing a kind of hierarchical structure. The modular architecture described can be compared to a tree; the root node being the recurrent neural network, and the leaves being the expert phonetic context single layer perceptron modules. This is similar to [5], where a TDNN was used a broad-class decision module to set of sub-networks of TDNNs.

Figure 4 shows this hierarchical structure in operation. The CI RNN stage operates in its normal fashion, generating frame-by-frame monophone posterior probabilities. At the same time the CD modules take the state vector generated by the RNN as input, in order to classify into a context class. The RNN posterior probability outputs are multiplied by the module outputs to form context-dependent posterior probability estimates. Thus the recurrent neural network can be envisaged as a *gating network* to the phonetic context layer, similar in operation to a mixture of experts model.

Hence, this structure is a two step classifier, firstly making a decision amongst the phone classes, and then deciding within a phone class. These modules were designed to extract useful and complementary information to the existing CI RNN stage, in a computationally inexpensive format.

Amendments to the NOWAY Decoder

No major amendments to the NOWAY decoder were necessary for it to run with more posterior probability inputs. However, there were pruning issues which were dependent on the phone set size, but are too detailed for this report. The same pruning parameters used for context-independent decoding were used for context-dependent decoding.

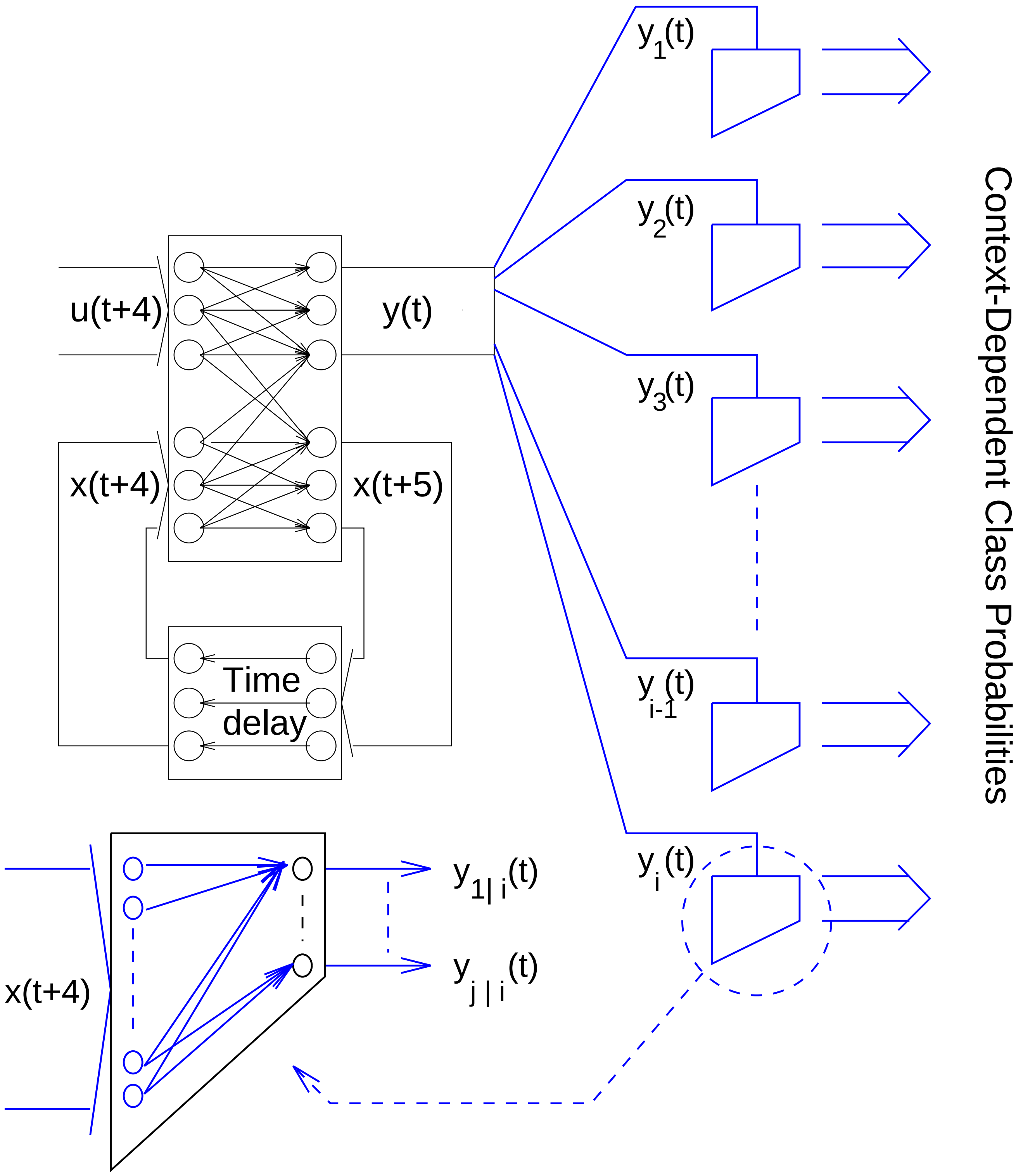

Figure 4: The phonetic context-dependent RNN modular system.

6 Evaluation of the Context-Dependent System

The context-independent network was trained on the ARPA Wall Street Journal SI84 Corpus. The new phonetic context-dependent classes were clustered on the MEL+ representation of the acoustic data, according to the decision tree algorithm given in Section 2. The split score threshold, the minimum number of frames at a node and the maximum number of terminal nodes were altered to result in two context-dependent phone sets: one with 205 phones and the other with 527 phones. Running the data through the recurrent network in a feed-forward fashion to obtain three million frames with 256 dimension state vectors took approximately 8 hours (on an HP735 workstation). Training all the context-dependent networks on all the training data takes between 4–6 hours in total (on an HP735 workstation).

The context-dependent modules were cross-validated on the Spoke 5 Development Test Set to find the optimum number of training epochs for each module. The process of obtaining the phonetic context-dependent posterior probabilities (using equation 11) is simply given by multiplying the context-dependent module posteriors by their associated gating probability (given by the RNN),

$$y_{ij}(t) = y_i(t) y_{j|i}(t) \tag{12}$$

where $y_{ij}(t)$ is the phonetic context-dependent posterior probability of phone class i in context class j. Unfortunately this was found to give only a modest reduction in word error rate. This could be due to a combination of reasons. Firstly there is likely to be a dynamic range mismatch between the posteriors of the CI recurrent network and the CD modules. (This can be likened to the sort of mismatch between frame-by-frame posteriors of acoustic evidence and a statistic language model). Secondly when obtaining the state vectors for training, this assumes that, there is a perfect monophone Viterbi segmentation of the labelled data and that there is a perfect gating network. Hence, equation 12 was amended, such that there would exist an information scaling between the CI RNN and the CD modules. Note that this is then re-normalised so that the smoothed estimate of the context-dependent posteriors, $y^s_{ij}(t)$, sums to one.

$$\begin{aligned} y^s_{ij}(t) &= \frac{y_i(t)[y_{j|i}(t)]^\alpha}{\sum_i y_i(t) \sum_j [y_{j|i}(t)]^\alpha} \\ 0.0 \leq \alpha &\leq \infty \end{aligned} \tag{13}$$

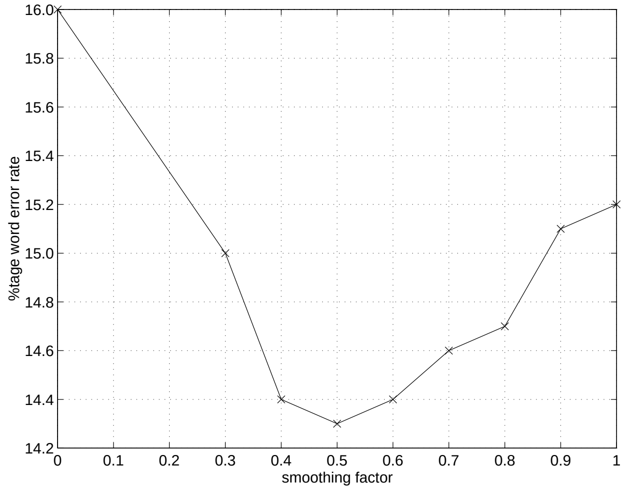

Figure 5: Word errors on the 1993 Spoke 5 development test set vs. the smoothing factor α, for the CD205 model set.

If α is less than one, then this "de-weights" the information content of the context modules $y_{j|i}(t)$. The value for the smoothing factor α was found empirically on a development set. An example of how the performance varies with α can be seen in Figure 5.

Test Sets	**CI System**	**CD205 System**		**CD527 System**	
	% WER	**% WER**	**% Red^n**	**% WER**	**% Red^n**
1993 Spoke 5 dev test	16.0	14.0	12.7	13.6	14.9
1993 Spoke 6 dev test	14.6	12.2	16.3	11.7	19.8
1993 eval test	15.7	14.3	8.4	13.7	12.6
Parameters	86800	137686		220944	

Table 3: Comparison of the context-independent (CI) system with the context-dependent systems (CD205 and CD527), for 5000 word, bigram language model tasks.

1995 Test Sets	**CI System % WER**	**CD System % WER**	**Red^n % WER**
US dev test	12.8	11.3	12.2
US eval test	14.3	12.9	9.8
UK dev test	15.6	12.7	18.9
UK eval test	16.5	13.8	16.3

Table 4: Comparison of the context-independent (CI) systems with the context-dependent systems (CD527US and CD465UK), for 20,000 word tasks. All tests used the 1993 standard trigram language model. The evaluation WER are official adjudicated results.

Results of the two context-dependent systems that were built, compared with the context-independent baseline are shown in Table 3, for the 5k Spoke 5 and 6 Development Test Sets and the 5k 1993 H2 C1 Evaluation. The CI System is a single MEL+ front-end RNN. The CD Systems augment this single MEL+ front-end RNN. The phone duration modelling simply used minimum duration [9]. The results in the table have optimally tuned parameters for both the baseline and the context-dependent system.

The context-dependent systems were also applied to larger tasks such as the 1995 SQUALE 20,000 word development and evaluation sets [20]. The American English context-dependent system (CD527) was extended to include a set of modules trained backwards in time (which were log-merged with the forward context), to augment a four way log-merged context-independent system [7]. A similar system was built for British English which used 465 context-dependent phones (CD465). Table 4 shows the improvement gained by using context models. The context-dependent systems shown achieved the lowest reported word error rate for both languages [20].

As a result of improved phonetic modelling the search space was reduced, resulting in faster decode speed for the NOWAY decoder, even though there were roughly ten times as many context-dependent phones compared to the monophones. This is highlighted in Table 5.

Test	**CI** **Utterance Av.** **Decode Speed (s)**	**CD** **Utterance Av.** **Decode Speed (s)**	**Speedup**
American English	67	31	2.16
British English	131	48	2.73

Table 5: Comparison of average utterance decode speed of the context-independent (CI) systems with the context-dependent systems (CD527US and CD465UK), for 20,000 word evaluation tasks. All tests use a trigram language model. All tests used the same pruning levels.

7 Conclusions

7.1 Summary

This report has outlined the construction of a very effective phonetic context-dependent system for the recurrent neural network framework. The various issues addressed have been:

- How to obtain context-dependent observation probabilities in terms of posterior probabilities as computed by the RNN and the context modules.
- How to decide upon the phonetic context-dependent classes. Section 2 uses a decision-tree clustering algorithm for this purpose.
- How to apply a simple architecture. This was necessary so that a system could be built swiftly and "bolted" on top of existing CI RNNs. A modular approach was taken to the design of such an architecture.
- How to achieve fast and effective training of the context modules. This was done by training on the state vector which was assumed to contain all the contextual information necessary for good context-class discrimination. Since the state vector is split into a large number of hyper-planes, it is assumed that the context classes are linearly separable. Hence, training was done with a single layer perceptron trained using a gradient descent technique.

7.2 Discussion

The report has discussed a successful way of integrating phonetic context-dependent classes into the current ABBOT hybrid system. The architecture followed a modular approach which could be used to augment any current connectionist-HMM hybrid system. Fast training of the context-dependent modules was achieved. Training on all of the SI84 corpus took between 4 and 6 hours. Utterance decoding was performed using the standard NOWAY decoder (with some minor modifications). Decoding speed of the context system was over twice as fast as the baseline system (for 20,000 word tasks). The results in Table 3 and Table 4 suggests that the phonetic context-dependent modelling can improve performance over the context-independent system, although the improvement witnessed is not nearly as great as that seen by equivalent HMM systems (Table 1). This is most likely due to the fact that:

- The internal acoustic context modelling in the recurrent network is better than that of the CI HMM systems. Hence, the baseline used for improvement already contains significant context modelling.
- The context modules are not as well trained as they could be. Merging context modules trained on the state vectors from a forward and a backwards through time RNN, is expected to give an improvement in performance at the context module level. This expectation is not without basis. In a standard "forward through time" RNN, the state vector contains left

context information due to the network's recurrent nature, and right context information due to the four frame delay. This contextual state vector information has different dynamics for the RNN "trained backwards through time", since the left and right context information sources have been reversed. This feature is likely to be well exploited when training "forwards" and "backwards" context modules.

- There are not as many context dependent classes in this system as there are in conventional context-dependent HMM systems.

The reduction in error rate is carried across all test sets, different size vocabularies (and language models) and languages (ie British English). Even though the reduction in error rate is not as large as that seen in the CD HMM systems, this is still a significant and consistent improvement. This has also been attained without a dramatic increase in the number of parameters, and still has orders of magnitude fewer parameters than context-dependent HMM systems.

References

[1] H. Bourlard and N. Morgan. Continuous Speech Recognition by Connectionist Statistical Methods. *IEEE Transactions on Neural Networks*, 4(6):893–909, November 1993.

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

[3] Breiman, Friedman, Olshen, and Stone. *Classification and Regression Trees.* Wadsworth & Brooks, 1984.

[4] M. Cohen, H. Franco, N Morgan, D Rumelhart, and V.Abrash. Context-Dependent Multiple Distribution Phonetic Modeling with MLPs. In *NIPS5*, 1992.

[5] L. Devillers and C. Dugast. Hybrid System Combining Expert-TDNNs and HMMs for Continuous Speech Recognition. In *ICASSP*, volume II, page 165, 1994.

[6] 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*, (8):211–222, 1994.

[7] M.M. Hochberg, G.D. Cook, S.J. Renals, and A.J. Robinson. Connectionist Model Combination for Large Vocabulary Speech Recognition. In *Neural Networks for Signal Processing*, volume IV, pages 269–278, 1994.

[8] M.M. Hochberg, G.D. Cook, S.J. Renals, A.J. Robinson, and R.S. Schechtman. The 1994 ABBOT Hybrid Connectionist-HMM Large-Vocabulary Recognition System. In *Spoken Language Systems Technology Workshop*, pages 170–6. ARPA, January 1995.

[9] M.M. Hochberg, S.J. Renals, A.J. Robinson, and D.J. Kershaw. Large Vocabulary Continuous Speech Recognition Using a Hybrid Connectionist-HMM System. In *ICSLP*, volume III, pages 1499–1502, 1994.

[10] M.I. Jordan and R.A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6:181–214, 1994.

[11] A. Kannan, M. Ostendorf, and J.R. Rohlicek. Maximum Likelihood Clustering of Gaussians for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 2, No. 3:453–5, July 1994.

[12] P. Kohn and J. Bilmes. Ring Array Processor (RAP): Software Users Manual Version 1.0. Technical Report TR-90-049, International Computer Science Institute, October 1990.

[13] F. Kubala. Design of the 1994 CSR Benchmark Tests. In *Spoken Language Systems Technology Workshop*, pages 41–6. ARPA, January 1995.

[14] P. Ladefoged. *A Course in Phonetics.* Harcourt, Brace and Jovanovich, 1982.

[15] Kai-Fu Lee. *Automatic Speech Recognition; The Development of the SPHINX System.* Kluwer Acedemic Publishers, 1989.

[16] J.D. O'Connor. *Phonetics.* Pelican Books, 1986.

[17] A.J. Robinson. An Application of Recurrent Nets to Phone Probability Estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305, March 1994.

[18] A.J. Robinson, H.Bourlard, M. Hochberg, D. Kershaw, N. Morgan, and S. Renals et al. A Neural Network Based Speaker Independent , Large Vocabulary, Continuous Speech Recognition System: The WERNICKE Project. In *EuroSpeech*, volume III, pages 1941–1944, 1993.

[19] Tony Robinson, Mike Hochberg, and Steve Renals. The Use of Recurrent Networks in Continuous Speech Recognition. In C.H. Lee, K.K. Paliwal, and F.K. Soong, editors, *Automatic Speech and Speaker Recognition - Advanced Topics*, chapter 19. Kluwer Academic Publishers, 1995.

[20] H.J.M. Steeneken and D.A. Van Leeuwen. Speech-Recognition Systems: The SQALE Project (Speech Recognition Quality Assessment for Language Engineering). To Appear in Eurospeech, September 1995.

[21] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics,Speech and Signal Processing*, 37(3):328–39, March 1989.

[22] P.C. Woodland and S.J. Young. The HTK Tied-State Continuous Speech Recogniser. In *Eurospeech*, volume 3, pages 2207–10, 1993.

[23] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-Based State Tying for High Accuracy Acoustic Modelling. *Spoken Language Systems Technology Workshop*, January 1994.

[24] G. Zavaliagkos, Y. Zhoa, R. Schwartz, and J. Makhoul. A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 2(1, Part II):151–9, January 1994.

[25] Y. Zhoa, R. Schwartz, J. Sroka, and J. Makhoul. Hierarchical Mixtures of Experts Methodology Applied to Continuous Speech Recognition. In *ICASSP*, volume 5, pages 3443–6, May 1995.

A Example Question Set for Tree Clustering

Question	**Phones in Question Grouping**
Class-Stop	b d dx g k p t
Class-Closure	bcl dcl gcl kcl pcl tcl
Class-Nasal	en em m n ng
Class-Fric	ch dh f jh s sh th v z zh
Class-Liquid	el hh hv l r w y
Class-Glide	l r w y
Class-Vowel	eh ih ao ae aa ah uw uh er ay oy ey iy aw ow ax axr ix
Coronal	ch d dh dx el en jh l n r s sh t th zh
Non-Coronal	b em f g hh hv k m ng p v w y
Anterior	b d dh dx el em en f l m p s t th v w z
Non-Anterior	ch g hh hv jh k ng r sh y zh
Continuant	dh dx el em en f hh hv l m n ng r s sh th v w y z zh
Non-Continuant	b ch d g jh k p t
Alveolar	d dx en l n r s t z
Palato-Alveolar	sh zh
Velar	g k ng
Vowel-Front	ae eh ih iy
Vowel-Central	ah ax axr ix er ow
Vowel-Back	aa ao uh uw
Dipthong	aw ay ey oy
Vowel-High	ih ix iy uh uw
Vowel-Medium	ax axr eh er ow
Vowel-Lo	aa ae ah ao
Fortis	ch f k p s sh t th
Lenis	b d dh dx g jh v z zh
Fricative	dh f s sh th v z zh
Voiced-Stop	b d dx g
Unvoiced-Stop	k p t
Voiced-Closure	bcl dcl gcl
Unvoiced-Closure	kcl pcl tcl
Labial-Stop	b p
Alveolar-Stop	d dx t
Specific Phonemes As Question Classes	
sil	sil
aa	aa
. .	. .
zh	zh

Table 6: Decision-tree question set for the LIMSI-ICSI phone set