THE 1994 ABBOT HYBRID CONNECTIONIST-HMM LARGE-VOCABULARY RECOGNITION SYSTEM

M. M. Hochberg[†], *G. D. Cook*[†], *S. J. Renals*[‡], *A. J. Robinson*[†] *& R. S. Schechtman*[†]

[†]Cambridge University Engineering Department, Cambridge CB2 1PZ, England
[‡]Department of Computer Science, University of Sheffield, Sheffield S1 4DP, England

ABSTRACT

ABBOT is the hybrid connectionist-hidden Markov model large-vocabulary speech recognition system developed at Cambridge University. In this system, a recurrent network maps each acoustic vector to an estimate of the posterior probabilities of the phone classes. The maximum likelihood word string is then extracted using Markov models. As in traditional hidden Markov models, the Markov process is used to model the lexical and language model constraints. This paper describes the system which participated in the November 1994 ARPA evaluation of continuous speech recognition systems. The emphasis of the paper is on the differences between the 1993 and 1994 versions of the ABBOT system. This includes the utilization of a larger training corpus (SI284 versus SI84), the extension of the lexicon from 5,000 words to 65,000 words, the application of a trigram language model, and the development of a near-realtime single-pass decoder well suited for the hybrid approach. Experimental results are reported for various test and development sets from the November 1992, 1993 and 1994 ARPA benchmark tests.

1. INTRODUCTION

The hybrid connectionist-hidden Markov model approach uses an underlying hidden Markov process to model the time-varying nature of the speech signal and a connectionist system to estimate the observation likelihoods within the hidden Markov model (HMM) framework [1]. ABBOT is a large-vocabulary speech recognition system based on the hybrid approach and utilizes a recurrent network for acoustic modeling. The major advantage of this approach is that the recurrent network acts a nonparametric model that is able to capture temporal acoustic context. Subsequently, the ABBOT system is a able to achieve very good performance using context-independent phone models.

The ABBOT system has participated in the 1993 [2] and 1994 ARPA continuous speech recognition (CSR) evaluations. This paper provides a basic description of the 1994 system and reports on the improvements made to the system over the past year. The acoustic modeling used for the 1994 evaluations is presented in the following section. This section describes the talker-cluster-based approach taken for extending the ABBOT training from the SI-84 to the SI-284 database. Section 3 briefly presents the application of a phone deletion penalty term to the decoding process. For the 1994 H1:P0 task, ABBOT was extended to handle a 65,532 word vocabulary with a trigram language model and the details are given in section 4. The ABBOT decoder – described in section 5 – takes advantage of the properties of the connectionist acoustic model to dramatically reduce the recognition search time. Section 6 reports on the performance of the ABBOT system on various ARPA CSR development and evaluation tasks.

2. ACOUSTIC MODEL

This section describes the acoustic modeling process used in the ABBOT system. This includes a brief description of the front-end, structure of the observation model (i.e., the recurrent network), and the training process used for estimating the parameters of the connectionist component.

2.1. Acoustic Feature Representation

Two sets of acoustic features are used by the ABBOT system; MEL+ – a 20 channel mel-scaled filter bank with three voicing features [3], and PLP – 12th order cepstral coefficients derived using perceptual linear prediction and log energy [4]. Both sets of features were computed from 32 msec windows of the speech waveform every 16 msec. Note that the choice of frame rate was determined for performance maximization not for decoding speed. To increase the robustness of the system to environmental conditions, the statistics of each feature channel were normalized to zero mean with unit variance over each sentence. To reduce the storage requirement, each feature channel at each frame was coded into a single byte.

The MEL+ and PLP feature vectors are represented in different fashions at the input to the connectionist probability estimator. The recurrent network builds up a representation of the past acoustic context which implies the ordering of the input data is important. The ABBOT system utilizes recurrent networks trained using forward- and backward-in-time input sequences of both MEL+ and PLP feature vectors. Additional acoustic context is encoded into the network inputs by augmenting the feature vectors with either adjacent frames or estimates of the feature derivatives.

2.2. Recurrent Network Structure

The basic acoustic modeling system is illustrated in figure 1 and fully described in [5, 6]. For each input frame, an acoustic vector, $\mathbf{u}(t)$, is presented at the input to the network along with the current state, $\mathbf{x}(t)$. These two vectors are passed through a standard single layer, feed-forward network to give the output vector, $\mathbf{y}(t-4)$, and the next state vector, $\mathbf{x}(t+1)$. The sigmoid and softmax nonlinearities are applied to the state and output nodes, respectively. The output vector represents an estimate of the posterior probability of each of the phone classes, i.e.,

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

where $q_i(t)$ is state i at time t and $\mathbf{u}_1^t = \{\mathbf{u}(1), ..., \mathbf{u}(t)\}$ is the input from time 1 to t. The output is delayed by four frames to account for forward acoustic context. The state vector provides the mechanism for modeling context and the dynamics of the acoustic signal. There is one output node per phone and the recurrent network generates *all* the phone probabilities in parallel.

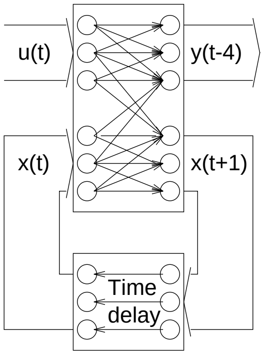

Figure 1: The recurrent network used for phone probability estimation.

The acoustic training procedure is fully described in [5, 6]. The approach is based on Viterbi training where each frame of training data is assigned a phone label based on an utterance orthography and the current model. The recurrent network is then trained – using the back-propagation-through-time algorithm [7] – to map the input acoustic vector sequence to the phone label sequence. The labels are then reassigned and the process iterates. The initial alignments for the ABBOT system were derived from a recurrent network trained on the TIMIT database.

2.3. Front-end Model Merging

The ABBOT results from the 1993 ARPA evaluations showed a dramatic performance improvement from merging multiple recurrent networks trained on different input representations. A linear merging approach was used for the 1993 system, i.e., setting

$$y_i(t) = \frac{1}{K}\sum_{k=1}^{K} y_i^{(k)}(t) \tag{2}$$

where $y_i^{(k)}(t)$ is the posterior probability estimate by the kth model. Recent work [8] has indicated that a better approach is to merge the network outputs in the log domain, i.e.,

$$\log y_i(t) = \frac{1}{K}\sum_{k=1}^{K} \log y_i^{(k)}(t) - B \tag{3}$$

where B is a constant to insure that $\mathbf{y}$ is a valid probability distribution. The log-domain merge is the approach used by the 1994 ABBOT system.

2.4. Talker-Cluster Merging

Recent work on merging networks trained on different talkers has been motivated by two factors. The primary goal was to fully utilize the great amount of training data available. Due to memory and time limitations, it was difficult to directly train a recurrent network using the full SI-284 training corpus. The approach taken was to use multiple networks trained from subsets of the data and merge the outputs. The second motivating factor for this approach was to reduce the effects of inter-speaker variability. To minimize this effect, multiple connectionist models are each trained on a subset of the training data. The subsets are formed by clustering the utterances so as to minimize the variability within each subset.

Talker Clustering The talker clustering is based on the LBG algorithm for vector quantization [9] using a distance measure which has been shown to give good discrimination performance for speaker identification [10]. The clustering algorithm is a hierarchical divisive procedure and starts with a single cluster consisting of a pattern vector for each training utterance. The pattern vector is the sample covariance matrix S of the input data from the utterance. Each cluster j is described by a cluster covariance matrix Σ_j derived from the utterances in the cluster. The data is randomly split into two disjoint clusters and the cluster covariances are re-computed. Each pattern is assigned to a cluster based on estimate of the log-likelihood of S given Σ_j, i.e.,

$$l(S;\Sigma_j) = \frac{Nn}{2}\left\{\log(|\Sigma_j^{-1}S|)^{1/n} - \frac{1}{n}\mathrm{tr}(\Sigma_j^{-1}S)\right\}. \tag{4}$$

Here, N is the number of samples in the utterance, n is the dimensionality of the feature vector and tr indicates the trace of a matrix. The cluster covariances are re-computed after all of the patterns have been assigned to a cluster. This process continues until each cluster is stable and there is no movement of patterns between clusters. The cluster consisting of the largest number of patterns is then randomly split into two clusters, and the process continues as before. This continues until the desired number of clusters – five in the experiments described here – have been created.

For reasons related to training of the recurrent networks, it is desirable to have the same number of tokens in each cluster. This is accomplished by assigning a scale factor β_j to each cluster log-likelihood score. This subset-size normalization is applied after completion of the clustering algorithm. The clustering procedure is then re-run using fixed cluster covariances and only re-assigning the utterance labels. This is an iterative procedure where β_j is defined as

$$\beta_j^m = \left\{1 + \left(\frac{n_j - N}{N}\right)\epsilon\right\}\beta_j^{m-1} \tag{5}$$

and where n_j is the number of patterns in cluster j, N is the desired number of patterns per cluster, m is the iteration, and ϵ is a small constant.

Merging Based on Talker Cluster Each cluster is defined in terms of its covariance Σ_j, its weight β_j and a list of utterances U_j which generate the covariance. The U_j are used as training data for cluster dependent models. Thus, for each subset of the data, a recurrent network is trained to estimate phone probabilities. When an utterance is to be decoded, the covariance of the acoustic feature vectors S is computed. The posterior probability of the jth cluster model ω_j given the data U is then estimated by

$$P(\omega_j|U,\alpha) \sim l(S;\Sigma_j)^{\alpha}\beta_j^{\alpha} \tag{6}$$

where α is an empirically determined tuning parameter. The outputs of the recurrent networks are then merged using

$$y_i(t) = \sum_{k=1}^{K} P(\omega_k|U,\alpha)\, y_i^{(k)}(t). \tag{7}$$

The 1994 ABBOT system utilized both front-end and talker-cluster merging. With four different feature representations and five talker clusters, this resulted in training and decoding with twenty recurrent networks with approximately two million free acoustic modeling parameters. Generating the output probabilities was accomplished by first merging across cluster for each feature representation and then log-domain merging across each feature representation. The computational requirement for the recognition-time probability estimation process can be significantly reduced by a factor of five by only using the most probable cluster (i.e., setting $\alpha = \infty$) with only a minimal impact on performance.

3. PHONE DELETION PENALTY

During the investigation of different phone-duration models, it was discovered that providing a *phone deletion penalty* could substantially improve the recognition performance [11]. The phone deletion penalty is a multiplicative scale factor to the state path likelihood, i.e., the applied state path likelihood is given by

$$L^{\mathrm{app}}(\mathrm{path}) = \kappa^{n} L(\mathrm{path}) \quad (8)$$

where $L(\mathrm{path})$ is the path likelihood specified by the Markov process on the states, κ is the phone deletion penalty, and n is the number of phones in the path. For $\kappa > 1$, this approach discourages deletions. This value was determined empirically on development data.

4. LEXICON AND LANGUAGE MODEL

Three different word lists and four different language models were used in the experiments reported in this paper. The standard word lists used were the 5k word, closed vocabulary specified for the 1993 H2:C1 test, and the 20k word, open vocabulary specified for the 1994 H1:C1 test. A lexicon with 65,532 entries was also developed. The word list was generated from the complete set of WSJ text data, including the material used as development data. The procedure for determining the word list is as follows:

1. Every year of WSJ text data (excluding the development test data) was split into two, and unigram count files were derived for the fifteen periods from the start of 1987 to the first half of 1994.
2. Each unigram count file was converted to a unigram p.d.f. then weighted by α^t where $0.0 < \alpha \leq 1.0$ and t was the number of six month periods before the first half of 1994. The result was summed and then renormalised to a p.d.f. The word frequency list was examined and a list of 1,526 non-words was created to be excluded from the lexicon. These were mostly misspellings, acronyms and formatting errors. The most probable 65,536 words not in the excluded list were extracted and the out-of-vocabulary (OOV) rate computed for the development test set.
3. Step 2. was repeated to find the α that minimised the OOV rate.
4. Steps 1. and 2. were repeated with the fixed α and the development test material included. The most frequent 65,532 words formed the final lexicon.

This goal of this procedure is to discount those words that did not occur recently.

The 1993 ABBOT system used a pronunciation lexicon supplied by Dragon Systems [12]. This lexicon provided coverage for the SI-284 training and 1993 CSR evaluations. The corresponding phone set uses 79 phone symbols where the vowels have three levels of stress. Because of the new requirements for larger and different vocabularies, the 1994 ABBOT system employed pronunciations derived primarily from a lexicon supplied by LIMSI-CNRS. The LIMSI-CNRS lexicon did not provide full coverage of the 65,532 words specified in section 4 and was expanded in the following fashion. First, the 1993 LIMSI dictionary was extended with those words that could easily be derived from the existing entries by the addition of suffices or the merging of two words. This resulted in approximately 35,000 entries. The remaining entries were obtained by the International Computer Science Institute (ICSI) from the CMU, COMLEX, TIMIT, OGI Numbers, and BEEP dictionaries and a TTS system using a probabilistic mapping technique to unify the phone sets and provide multiple pronunciations [13]. A set of rules provided by ICSI were employed to expand the phone set/lexicon to specify stops as flaps or closures and (possibly) releases [13]. This resulted in a total of 54 phones in the lexicon. Multiple pronunciations were assigned probabilities based on their frequency of occurance in the training data [2].

The standard language models used were the 1992 20k, open, non-verbalized punctuation trigram, the 1993 H2:C1 bigram, 1993 H2:P0 trigram, and the 1994 H1:C1 trigram language models. For the 65,532 word list, the CMU language modeling tools were used to build a standard 1-3 backoff trigram language model. This language model was based on the counts of both the full training and development test sets. Note that standard text processing was used without modifications to the text filters.

5. DECODER

The acoustic modeling in ABBOT is somewhat different to the context-dependent mixture model approach used in most other systems. In particular, the following differences have proven to be important in the design of an efficient decoder:

- The connectionist system directly estimates posterior probabilities, $P(\mathrm{phone} \mid \mathrm{data})$, rather than likelihoods, $p(\mathrm{data} \mid \mathrm{phone})$;
- Context-independent acoustic modeling leads to a small set of basic HMMs (typically 40–80), rather than several thousand context-dependent models;
- Connectionist probability estimation enables the computation of all phone probability estimates at each frame without much additional computational cost.

By making effective use of these properties of hybrid systems, the single-pass decoder – referred to as NOWAY – operates at approximately 15× realtime on an HP735 workstation for a 20,000 word task using a backed-off trigram language model. At the cost of 7% relative search error, decoding time can be speeded up to approximately realtime.

5.1. Basic Algorithm

The search algorithm described here is partially time-asynchronous and is based on the ideas of stack decoding [14, 15]. The Viterbi criterion is used — i.e., the full likelihood is not computed — so the algorithm may be regarded as a reordered time-synchronous Viterbi search. For simplicity of presentation, we consider decoding a single utterance of length T.

The basic data structure of the search algorithm is a priority queue, or *stack*. The elements of the stack are *hypotheses*; a hypothesis h contains a proposed decoding W_h up to a given reference time t_h with a log likelihood L_h. W_h is comprised of a word sequence $\{w_h(0), w_h(1), \ldots\}$.

A fundamental decision that must be made in the stack decoding algorithm is which hypothesis in the stack should extended. To avoid using a multi-pass or look-ahead approach, an approximation to the A^* criterion is used which utilizes an estimate of the least upper bound on the likelihood of all paths at a particular time. Using this approximation, hypotheses need only be compared with other hypotheses with the same reference time. This implies using a set of stacks: one for each time frame of the utterance to be decoded. This approach has been successfully used by Bahl and Jelinek [14] and Paul [15]. In NOWAY, an initial estimate of lub$L(t_h)$ (the least upper bound at time t_h) is generated from the network outputs. The n most probable phone posteriors (not including the most probable) are averaged and converted to a scaled likelihood by dividing by a uniform prior. This estimate of lub$L(t_h)$ is then updated whenever a particular hypothesis extension has a higher likelihood at t_h.

The bulk of the work is done when propagating the active hypotheses forward in parallel. For efficiency, the lexicon is stored as a tree. Each *node* in the tree corresponds to a phone in a particular set of pronunciations and the use of this structure reduces the number of constituent phone models required by a factor of three or four. The root node of the tree corresponds to a pause model — a single state silence model which may be skipped — to allow for optional inter-word pauses. The set of hypotheses (with the same extension start-time) is propagated through the same tree and share acoustic information with their scores differing only in language model information and start scores. The tree is searched in a time-synchronous, breadth-first manner, although there is no *a priori* reason for preferring this to a depth-first search.

The basic decoding algorithm can be summarized as follows:

```
1. Set $t = 0$; lub$L(\tau) = -\infty$ , $0 \leq \tau < T$; Initial null hypothesis:
   $t_h = 0$; $L_h = 0$ and $W_h = \mathtt{NULL}$.
2. Push initial hypothesis onto stack(0).
3. If (end-of-utterance) output top of stack(t) and exit.
4. Else process stack(t):
     • Pop all hypotheses into active hypothesis list, hlist.
     • If hlist is not empty expand hypotheses in parallel:
         – Activate root node of lexical tree
         – Propagate hypotheses forward time-synchronously
           and activate new nodes
         – Prune active nodes according to likelihood-based
           and posterior-based pruning criteria (see section )
         – Update lub$L(t)$ if required
         – At word-end nodes within envelope, extend hypotheses
           by one word, incorporate exact language model (LM)
           score, push hypotheses onto relevant stack.
         – Continue if any nodes are active
5. $t \leftarrow t + 1$; goto 3
```

The use of a language model is essential to constrain the search space in large vocabulary recognition. However there is a tradeoff between accessing the required language model probabilities and the efficiency gain obtained in the search by their application. Incremental caching of language model probabilities as they are accessed is used to aid efficient retrieval. In a tree-based lexicon, the correct way to take advantage of the language model to reduce the search is by computing the maximum language model probability for each node. This involves taking a maximum over the language model probabilities of all words that use that node in their pronunciation given the hypotheses that they are extending. This involves a significant amount of computation, particularly in nodes close to the root of the tree which are part of the pronunciations of many words. In these cases, an approximated upper bound on the language model probability – namely the maximum bigram probability given a context – is used instead of the exact value. The set of default bigrams is computed in advance and stored in a table. Experiments have indicated that using this approximation is more efficient at all word-internal stages of the search and the exact language model probabilities are used only at word ends. Incremental language model caching is still used at word ends, giving a 50% speedup. In this case, all hypotheses are propagated in parallel and only individually evaluated at word ends. Experiments in which individual hypotheses may be separately pruned at each node have been carried out, but do not show any efficiency improvements.

5.2. Pruning

In conventional and hybrid HMM systems, the search space is evaluated by computing likelihood estimates of the acoustic data having been generated by a particular utterance model. Pruning strategies are generally likelihood-based and involve the specification of a *envelope* Δ around the likelihood L of the most probable partial hypothesis at time t. Hypotheses whose likelihood falls outside the envelope (i.e., those hypotheses with a likelihood $L^{'} < L - \Delta$) are pruned. The number of hypotheses on the stack is also limited which provides another mechanism for reducing the search space. Both these likelihood-based pruning strategies are used in the ABBOT system.

Another NOWAY pruning strategy makes use of the phone posterior probabilities estimated by the connectionist system. These probabilities may be regarded as a local estimate of the presence of a phone at a particular time frame. If the posterior probability estimate of a phone given a frame of acoustic data is below a threshold, then all words containing that phone at that time frame may be pruned. This strategy can be efficiently implemented within a tree structured lexicon and is referred to as *phone deactivation pruning*. The posterior probability threshold used to make the pruning decision is empirically determined using development data and is constant for all phones. Phone deactivation pruning takes advantage of the fact that ABBOT's basic acoustic component directly estimates posterior probabilities rather than likelihoods. To carry out an equivalent approach in a likelihood-based system requires summing over a (possibly large) set of baseform HMMs.

		Acoustic	Input
System ID	Lexicon	Training	Format
1	Dragon	SI-84	5 frame window
2	LIMSI	SI-84	Δ parameters
3	LIMSI	SI-284	Δ parameters

Table 1: Summary of systems. The *input format* column refers to what information is presented at the input of the recurrent network and Δ indicates that differenced parameters are used.

6. RESULTS

Results are reported on evaluation and development tests from the 1992, 1993 and 1994 ARPA CSR evaluations. The particular tests evaluated in this paper are:

20k(92)	The 1992 evaluation test using an open 20,000 word vocabulary with non-verbalized punctuation.
S5(93)	The 1993 spoke 5 development test using a 5,000 word, closed vocabulary (Sennheiser microphone).
H2(93)	The 1993 small-site hub 2 evaluation task using a 5,000 word, closed vocabulary.
H1(94)	The 1994 hub 1 evaluation task using an unlimited vocabulary.

Various systems – reflecting the progression of ABBOT from the 1993 to 1994 ARPA CSR evaluations – were evaluated on the above tests. These systems are summarized in table 1. Please note that all the results reported here are not phone-mediated and the 1994 results are pre-adjudication. Unless otherwise noted, systems trained on SI-84 utilized front-end merging while systems trained on SI-284 utilized both front-end and talker-cluster merging.

6.1. Acoustic Modeling Results

Table 2 shows the performance of the two SI-84 systems on the H2(93) task. The top of the table shows the performance of the system using various single recurrent networks as the acoustic model (i.e., no merging). Here F- and B- indicate forward- and backward-in-time input to the recurrent network, respectively. The bottom rows of the table show the performance for a linear merge, log-domain merge, and log-domain merge with the phone deletion penalty.

Table 3 reports the ABBOT results for going from SI-84 to SI-284 training. Although the improvement is significant, it is not as great as seen by conventional HMM-based systems. This is probably due to insufficient training and improvement is expected with further investigation into the training schedule.

The results for different methods of talker-cluster merging are shown in table 4. The table shows that using only the most probable talker cluster has no real adverse affect on the performance. The table also shows the performance improvement gained by expanding the lexicon to 65,532 words. The difference in performance reflects the difference in OOV words between the two lexicons.

	Error Rate, %	
Feature	System 1	System 2
f-MEL+	16.2	15.5
b-MEL+	16.1	15.7
f-PLP	16.5	16.1
b-PLP	15.2	15.3
linear merge	13.4†	13.7
log merge	12.4	13.0
log merge + phone del. pen.	10.9	12.1

Table 2: Results showing the performance improvement for merging and phone deletion on the H2(93) task. Both systems utilize the standard bigram language model. A version of ABBOT close to the 1993 CSR H2:C1 evaluation system is denoted by †.

	Error Rate, %		
Test	System 2	System 3	Improv., %
H2(93)	9.3	8.1	13
H1(94)	17.5	14.7†	16

Table 3: Comparison of SI-84 and SI-284 acoustic training. System 2 and 3 reflect SI-84 and SI-284 training, respectively. Both systems use the standard 1993 5k vocabulary and trigram language model for the H2(93) test and the standard 1994 20k vocabulary and trigram language model for the H1(94) task. The 1994 CSR H1:C1 evaluation system is denoted by †.

Talker Cluster	Lexicon	Error Rate, %
merged	20k	14.7†
most probable	20k	14.8
merged	64k	12.9‡

Table 4: Performance for using talker-cluster merging and 65,532 word lexicon evaluated on the H1(94) with system 3. The 1994 CSR H1:C1 and H1:P0 evaluation systems are denoted by † and ‡, respectively.

6.2. Decoder Results

Table 5 illustrates the NOWAY decoding performance relative to the phone deactivation pruning threshold. Note that applying posterior-based pruning with a threshold of 0.000075 gives around an order of magnitude improvement in the decoding speed with an increased relative search error of less than 2%. The best parameter setting for realtime decoding is not shown in the table. However, using a posterior threshold of 0.0005, an envelope of 8 and a stack size of 7 results in realtime performance (on an HP735) with a relative search error of around 7%. On the H1:P0 task with 65,532 words and trigram language model, a decoding speed of 20 × realtime (HP735) was obtained with 2% relative search error (13.0% word error) using a posterior threshold of 0.000075, an envelope of 9 and a stack size of 15. As a final note on decoder performance, initial results indicate that the NOWAY decoder running on a PC (90MHz Pentium) with 64Mbytes of memory takes only twice as long as an HP735.

Pruning Parameters		S5(93)		20k(92)	
Envelope	Threshold	Time	Error	Time	Error
10	0. 0	165.3	12.2	175.1	12.4
10	0.000075	16.1	12.1	15.7	12.6
10	0.0005	4.3	12.2	3.9	12.9
10	0.003	1.4	14.3	1.3	14.9
8	0.0	46.8	12.5	50.4	12.6
8	0.000075	5.4	12.2	4.9	12.8
8	0.0005	1.7	12.6	1.5	13.6
8	0.003	0.6	15.0	0.6	15.8

Table 5: Decoding performance with respect to varying phone deactivation pruning threshold. The maximum stack size was set to be 31. In cases when the posterior-based pruning threshold was greater than 0.0, posterior-based pruning of leading silence was also employed.

7. SUMMARY

There are a few general conclusions which can be drawn about the 1994 ABBOT system.

- Log-domain merging and the application of a phone deletion penalty provide a simple, but effective means of improving the recognition performance.
- Although a significant improvement was gained by going from the SI-84 system to the SI-284 training, it was less than that reported by conventional HMM systems. Efficient use of the additional training data is still an important research area.
- The decode-time acoustic model computation can be reduced by 80% with only a minor degradation in recognition performance by using the most probable cluster.
- The development of the NOWAY decoder has been very successful. In particular, the extension to larger lexicons, the capability for long-span language models, and the use of phone deactivation pruning has resulted in a very powerful recognition system.

Considering that context-independent phone models are used for the sub-word HMMs, the performance of the ABBOT system – although certainly not the best – is quite good. Further work planned for the ABBOT system includes the continued investigation of talker-cluster merging approaches, training of the recurrent networks, application of context-dependent phone-duration models, the use of alternate input representations and the development of speaker-adaptation approaches.

8. ACKNOWLEDGMENTS

This work was partially funded by ESPRIT project 6487, WERNICKE. AJR and SJR were supported by EPSRC fellowships. The authors would like to acknowledge MIT Lincoln Laboratory and CMU for providing language models and associated tools, and Dragon Systems, LIMSI-CNRS, ICSI and Dave Pye for providing and/or helping with the pronunciation lexicons.

References

1. Bourlard, H. and Morgan, N. *Connectionist Speech Recognition: A Hybrid Approach*. The Kluwer International Series in Engineering and Computer Science. VLSI, Computer Architecture, and Digital Signal Processing. Kluwer Academic Publishers, Boston, Massachusetts, 1994.
2. Hochberg, M. M., Renals, S. J., and Robinson, A. J. "ABBOT: The CUED hybrid connectionist-HMM large-vocabulary recognition system." In *Proc. of Spoken Language Systems Technology Workshop*. ARPA, Mar. 1994.
3. Robinson, T. "Several improvements to a recurrent error propagation network phone recognition system." Technical Report CUED/F-INFENG/TR.82, Cambridge University Engineering Department, Sept. 1991.
4. Hermansky, H. "Perceptual linear predictive (PLP) analysis of speech." *J. Acoust. Soc. Am.*, 87:1738–1752, 1990.
5. Robinson, A. J. "An application of recurrent nets to phone probability estimation." *IEEE Transactions on Neural Networks*, 5(2):298–305, Mar. 1994.
6. Robinson, T., Hochberg, M., and Renals, S. "The use of recurrent neural networks in continuous speech recognition." In Lee, C. H., Paliwal, K. K., and Soong, F. K., editors, *Automatic Speech and Speaker Recognition – Advanced Topics*, chapter 19. Kluwer Academic Publishers, 1995.
7. Werbos, P. J. "Backpropagation through time: What it does and how to do it." *Proceedings of the IEEE*, 78(10):1550–1560, oct 1990.
8. Hochberg, M. M., Cook, G. D., Renals, S. J., and Robinson, A. J. "Connectionist model combination for large vocabulary speech recognition." In Vlontzos, J., Hwang, J.-N., and Wilson, E., editors, *Neural Networks for Signal Processing IV*, pp. 269–278. IEEE, 1994.
9. Linde, Y., Buzo, A., and Gray, R. "An algorithm for vector quantizer design." *IEEE Transactions on Communications*, COM-28(1):84–95, Jan. 1980.
10. Gish, H., Kransner, W., Russell, W., and Wolf, J. "Methods and experiments for text-indepedent speaker recognition over telephone channels." In *1986 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1986.
11. Hochberg, M. M., Renals, S. J., Robinson, A. J., and Kershaw, D. J. "Large vocabulary continuous speech recognition using a hybrid connectionist-HMM system." In *Proc. of ICSLP-94*, pp. 1499–1502, 1994.
12. Paul, D. B. and Baker, J. M. "The design for the Wall Street Journal-based CSR corpus." In *Proc. Fifth DARPA Speech and Natural Language Workshop*, pp. 357–362, Harriman, New York, Feb. 1992. DARPA, Morgan Kaufman Publishers, Inc.
13. Tajchman, G., Jurafsky, D., and Fosler, E. "Learning phonological rule probabilities from speech corpora with exploratory computational phonology." Submitted to ACL-95.
14. Bahl, L. R. and Jelinek, F. "Apparatus and method for determining a likely word sequence from labels generated by an acoustic processor." United States Patent, May 1988. Number 4,748670.
15. Paul, D. B. "An efficient A^* stack decoder algorithm for continuous speech recognition with a stochastic language model." In *1992 International Conference on Acoustics, Speech, and Signal Processing*, pp. 25–28, San Francisco, California, Mar. 1992. IEEE. Volume 1.