Proceedings of the Institute of Acoustics

Recent Developments in the HTK Large Vocabulary Continuous Speech Recognition System

J.J. Odell, V. Valtchev, P.C. Woodland & S.J. Young

Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, England.

1. INTRODUCTION

This paper describes recent experiments using the HTK large vocabulary continuous speech recognition system[6]. The system uses state-clustered mixture Gaussian cross-word triphone HMMs to allow an appropriate balance of acoustic modelling detail (model complexity) and parameter estimation accuracy for a given training corpus. The system decoder is able to integrate cross-word triphone acoustic models and trigram language models into a single recognition pass. The approach was evaluated in the ARPA November 1993 Wall Street Journal (WSJ) evaluation with both 5k word and 20k word vocabularies. The HTK based system had the lowest error rate reported on three of the four tests entered and the second lowest error rate on the fourth test.

Recently, our focus has been to optimise and further enhance the capabilities of the system. However experimentation using our original system was computationally costly. Therefore the decoder has been extended so that it can produce a network of recognition alternatives stored in a *word lattice* for each sentence. These word lattices can include different amounts of information. At their most detailed this allows the recovery of the exact sentence likelihood and segmentation for each hypothesis in the lattice and at the other extreme only specify a finite state syntax for constrained recognition. Use of these word lattices allows the computationally efficient optimisation of system parameters or evaluation of alternative acoustic and language models.

We have also ported our recognition technology to the Switchboard Database. This is a database of conversational telephone speech and has very different characteristics to the clean read speech of the WSJ corpus.

This paper first gives an overview of the complete HTK large vocabulary recognition system. The process of generating word lattices is described, as is the way in which they are used within the system. A number of recent experiments are then presented. These experiments include a comparison of alternative pronunciation dictionaries, tuning system parameters and the use of variable component mixture distributions. Finally results for the Switchboard corpus are described, and it is shown that the HTK large vocabulary system gives state-of-the-art performance on this task also.

2. SYSTEM OVERVIEW

The HTK large vocabulary system uses mixture Gaussian cross-word triphone HMMs, each with three emitting states. Speech data is coded using 12 MFCCs, normalised log energy, and the first and second differentials of these parameters. Cepstral mean normalisation is performed on a sentence by sentence basis.

The HMMs are built in a number of stages. First, using a pronunciation dictionary and sentence orthography, a phone level label string is generated by Viterbi alignment to choose the most likely pronunciation for each word. These labels are used to generate single Gaussian monophone HMMs, which are then cloned for every triphone context (ignoring word-boundaries) that occurs in the training data, and the parameters of the resulting single Gaussian cross-word triphone HMMs re-estimated.

To obtain good recognition performance, mixture Gaussian densities are required, but for the majority of triphone contexts there is insufficient data to accurately train the parameters of a mixture Gaussian model. Furthermore many of the cross-word triphones needed during decoding do not occur in the training data and a method for estimating these "unseen triphones" is needed. To solve both of these problems a tree-based state clustering procedure is used [5].

A phonetic decision tree is built for every monophone HMM state position to determine equivalence classes between sets of triphone contexts. The tree-growing procedure uses questions about the immediate phonetic context to repeatedly divide the states of triphones seen in training into groups. The final clusters contain triphone contexts that are acoustically similar but the tree growing algorithm ensures that they also have enough training observations for robust estimation of mixture Gaussian distributions. The output distributions of the members of each class are then tied to each other so that they share a single Gaussian distribution. Unseen triphones can be synthesised by using the decision trees to determine for each state which of these shared output distributions should be used. This allows a vocabulary independent system to be produced which allows unlimited vocabulary recognition.

The number of mixture components in each tied-state distribution is incremented using an iterative mixture-splitting and retraining procedure. A final optional stage in model building, clones this HMM set and re-estimates separate gender-dependent mean vectors, while retaining the gender independent variances. A more detailed description of the HMM build process is given in [4].

The system uses a time-synchronous one-pass decoder that is implemented using a dynamically built tree-structured network. This approach integrates the cross-word triphone acoustic models and either bigram or trigram language models directly into the search. The decoder saves computation and storage by using a tree-structured lexicon since much of the search effort is in the first phones of each word[2], but requires copies of the tree for differing acoustic and language model contexts. Details of the decoder architecture are given in [3].

3. WORD LATTICES

A word lattice forms a compact representation of many alternative sentence hypotheses. The lattices contain a set of nodes that correspond to particular time instants and arcs connecting these nodes that represent word hypotheses for the time period between two nodes. An example lattice is shown in Figure 1.

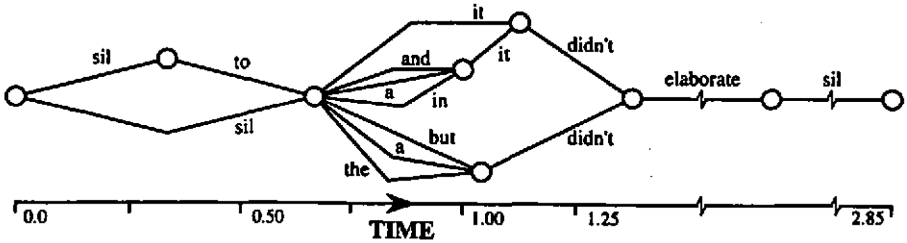

Figure 1: An example lattice.

Although the dynamic network decoder can operate in a single pass (which incorporates the most detailed acoustic and language models directly to provide constraints early in the search) this is still computationally expensive. In many experiments, for example during system development, a single test set will be decoded many times with similar systems. If this is the case, word lattices can be used to reduce the total computation required by allowing the use of computationally more efficient techniques to be used when a complete decode is not required.

Once these lattices have been constructed they can be used for a number of different purposes. Language model likelihoods can be scaled to optimise language model weight and/or word insertion penalties; a new language model can be applied using the same acoustic likelihoods and recognition performed using an A^* search through the lattice; and N-Best sentence hypotheses can be generated. Also, new acoustic models, and optionally a new language model, can be used with the lattice operating as a word-graph to constrain the search. In this case, the initial acoustic and language model likelihoods and word start/end times are not required. Of course a word lattice may also form the ideal interface to pass a hypothesis set to further stages of processing.

Typically each arc in the lattice will have both language model and acoustic model likelihoods associated with it and optionally a phonetic segmentation. Consequently the lattice will contain many copies of a single word since the acoustic likelihood of a word will depend on the surrounding phonetic context and the language model likelihoods will depend on the preceding words.

3.1 Lattice Generation

Lattice generation only requires minor modifications to the basic search strategy since multiple copies of each word are kept during the search and hypotheses only recombine at a small number of word end nodes. Information about which hypotheses recombine in the search after each frame are recorded and retained, but only the best hypothesis is extended. The multiple hypotheses can be recovered at each word-end node at the end of the utterance. This procedure will not generate exact solutions for any but the locally best path since it implicitly assumes that the start-time of each word is independent of all words before its immediate predecessor (the "word-pair approximation"). This procedure is similar to the lattice generation described in [1] except that here it has been extended to cross-word acoustic models and lattices can be generated with either bigram or trigram language models.

It has been found that the lattices generated in this manner can be pruned without adversely affecting lattice coverage. For each arc, the likelihood of the best complete path through the lattice which includes that arc is found. If this likelihood is more than a fixed threshold from the globally best path, then the arc is deleted. This pruning strategy can be efficiently implemented using an A* algorithm.

3.2 Lattice Error Rates

To measure the lattice quality, two lattice error rate measures are computed. The first determines whether a path corresponding to the true sentence exists in the lattice (lower bound on the sentence error rate), and the second is a lower bound on the word-error rate from rescoring the lattice, where the word-error rate is found by the usual string alignment procedure. Both measures are complicated by the possibility of out-of-vocabulary (OOV) words in the utterance. When these occur error bounds on the word error are computed; the lower bound assumes that any OOV words will be recognised correctly whilst the upper bound assumes they are never recognised.

Test Set	Lattice Density	Word Density	Sentence Error Rate	OOV Rate	Word Error Rate
Nov'92 5k	73	7	0.3	0.00	0.02
si_dt_05.odd	134	10	3.6	0.00	0.29
Nov'93 5k	135	9	6.5 - 10.7	0.29	0.42 - 0.73

Table 1: Lattice densities and % sentence/word error rates for 5k WSJ test sets and bigram.

Test Set	Lattice Density	Word Density	Sentence Error Rate	OOV Rate	Word Error Rate
Nov'92 64k	151	10	4.8 - 27.3	1.90	0.34 - 2.71
si_dt_20.odd	257	16	11.5 - 30.9	1.82	0.81 - 2.92
Nov'93 64k	284	14	8.0 - 28.2	1.71	0.64 - 2.76

Table 2: Lattice densities and % sentence/word error rates for 64k WSJ test sets and bigram.

The lattice sentence and word error rates for several WSJ 5k/64k test sets are shown in Tables 1 and 2. The set si_dt_20.odd contains alternate sentences from WSJ1 64k Hub development test data; si_dt_05.odd contains alternate sentences from the 5k Hub development test after sentences with OOV words were removed. The other test sets come from the November 1992 and 1993 evaluation tests. The lattices were generated with either a 5k or 20k bigram language model. The lattice density figure is the average number of lattice arcs per spoken word. As noted earlier, the lattices can contain many arcs for the same word, either with slightly different start/end times, with differing contexts, or different pronunciation variants. To account for these differences a second figure, the word density, which is the lattice density after a merging procedure which produces a word level finite state syntax in which different pronunciation and contexts have been merged.

Although there is some variability amongst test sets, the lattice word error rates for non-OOV words is less than 0.5% for 5k the test-sets and under 1% for the 20k data. These values imply that very similar error rates should be obtained for both rescoring the lattices and performing the full search and this has been confirmed in practice. For acoustic rescoring, the computation is reduced by approximately a factor of 20 by using the word lattices.

4. WSJ EXPERIMENTS

This section describes a number of recent experiments on the WSJ corpus. All model sets used the SI-284 training set. All tests used 64k WSJ test sets with the Lincoln Labs 1993 standard 20k open trigram language model.

4.1 Dictionary Comparison

Model Set	Nov'92	si_dt_20.odd	Nov'93
Dragon/GD	9.46	13.71	12.74
LIMSI/GI	10.28	13.32	12.57
LIMSI/GD	9.34	12.71	12.45

Table 3: % word error rates for different pronunciation dictionaries (64k test data)

The system that was built for the Nov'93 WSJ evaluation [4] used the Dragon Wall Street Journal Pronunciation Lexicon version 2.0 with some local modifications and additions. This system was compared to a similar one trained using pronunciations from the LIMSI 1993 WSJ Lexicon. The Dragon-based system contained 7,558 tied states and the LIMSI based system 7,299 tied states. Both systems used 10 component Gaussian mixtures. Results for the gender dependent (GD) version of the Dragon-based system and both gender independent (GI) and gender dependent models sets are given in Table 3 for different 64k WSJ test sets. Comparing the results for the GD systems, on average about a 4% improvement was obtained using the LIMSI dictionary. Furthermore, it can be seen that on average there is a 5% improvement from using GD models.

4.2 Word Insertion Penalty

The results reported in Table 3 use a language model weight (scale factor), but do not include a word insertion penalty. The word error rates with and without a word-insertion penalty are given in Table 4. These results show that on average there is a 3% reduction in word error rate. It was also found that for recognition of 5k closed vocabulary, there was no improvement using a word insertion penalty.

Penalty	Nov'92	si_dt_20.odd	Nov'93
N	9.34	12.71	12.45
Y	8.91	12.21	12.33

Table 4: % word error rates with a word-insertion penalty

4.3 Variable Component Mixtures

In the systems described above, all tied states have identical complexity. Varying the number of mixture components to more closely match the needs of acoustic variability and limited training data was investigated. The LIMSI based 10 component GI system (Const-10) was taken as a baseline. The number of mixture components in each state was reduced by finding the pair of components which could be merged such that the decrease in log likelihood on the training data was minimised. Provided that this decrease was less than a threshold, the selected components were merged and the procedure repeated. In addition, any mixture component with a total occupation count less than a second threshold was merged regardless. This procedure was used to reduce the total number of Gaussians in the system by 20%. After component merging, all models were re-estimated to form the Var-8 system. Table 5 shows the word error rates for two 64k test sets. For comparison purposes, results are also included for a system with 8 mixture components per state (Const-8) and a system (Var-10) with an average of 10 components per state formed by increasing the number of components in every state of the Var-8 system by two.

System	#Gaussians	si_dt_20.odd	Nov'93
Const-8	58,456	13.96	13.06
Const-10	73,070	13.32	12.57
Var-8	58,431	13.22	12.42
Var-10	73,045	12.95	12.65

Table 5: % word error rates for variable component mixture systems (Var-X). The total number of Gaussians in the HMM sets is also shown.

The mixture merging process results in a small improvement in performance coupled with a useful reduction in the number of parameters. However, when complexity is restored to give an average of 10 mixture components per state, the performance does not improve further. The conclusion of this preliminary experiment is that mixture component merging may be useful for reducing the number of parameters but it does not appear to lead to a significant increase in accuracy.

5. SWITCHBOARD EXPERIMENTS

The HTK large vocabulary recognition system has also been applied to recognition of speech from the telephone based Switchboard corpus. This is a large corpus of casual conversational speech on a variety of different topics. The subjects were prompted with an initial topic and the resulting conversation was encoded in 8-bit mu law at telephone bandwidth (8kHz sampling). A portion of the corpus used at the *1994 Frontiers in Speech Processing Workshop* held at Rutgers University was used for recogniser training and testing. The training set consisted of about 2.6 hours of speech data taken from 4 topics together with language model training data consisting of 2 million words from 70 topics. The results reported here are for the 10 topic development test set, defined for the workshop, that used a 4129 word vocabulary. The pronunciation dictionary used for training and testing was provided by BBN, as were the training and test orthographic transcriptions.

A state-clustered cross-word gender independent system was trained. The system used 1221 tied states with 6 component mixture distributions for each state. Bigram and trigram back-off language models were generated using the two million words of Switchboard transcriptions. The bigram had a perplexity of 115 measured on the dev-test data and the trigram model reduced this to 101. The small difference in perplexity between the bigram and trigram (compared to WSJ language models) was probably due to the relatively small amount of data available for constructing langauge models and the mis-match between training and testing topics. A *cheating* word-pair language model defined for the workshop was also used. This was constructed from the words in the complete acoustic training and test data and it had a perplexity of 51.8 on the development test data.

Grammar	Test Set	Perplexity	Lattice Density	Word Density	Sentence Error Rate	Word Error Rate
Word-pair	dev	51.8	1349	14	60.8	13.1
Bigram	dev	114.5	1332	24	65.0	11.6
Bigram	eval	101.7	1402	24	73.9	14.2

Table 6: Lattice densities and % lattice sentence/word error rates for Switchboard lattices.

Table 6 shows the size and error rates of lattices generated for the test data with bigram and word-pair language models. Despite tight lattice pruning these lattices are very large and this reflects the general difficulty of the search required because of the poor fit of the models. Although these lattices are very large and have a high word error rate, using them for constrained recognition still results in an order of magnitude reduction in the computation required and gives results which are within 0.5% of those obtained in a full search.

Taking the best path only through the lattice gave a 60.3% word error rate (i.e. bigram word error rate) which was reduced to 57.7% word error using the trigram. These figures represent state-of-the-art performance on this data set!

The acoustic training data was only a small part of the complete corpus (which is over 200 hours of speech) and our system was much smaller than or WSJ systems (10,000 Gausians as opposed to over 150,000 for our gender dependent system). We had found that for WSJ an increase in training data from 14Hrs to 66Hrs (from si84 to si284 training) had resulted in a 25% reduction in error rate. We expected to be able to improve our Switchboard as the quantity of the training data and hence size of system increased. Unfortunately the original data was unsuited to training a recogniser since the corpus was recorded in the form of long conversations. The HTK Baum-Welch re-estimation tool (HERest) requires sentence length files for efficient operation. Consequently an automatic procedure was used to split the conversations into shorter sentences and to verify the accuracy of the transcriptions. (This was necessary due to occasional gross errors in the transcription process, such as swapping the labels indicating which speaker had spoken each utterance).

This procedure produced over 40Hrs of speech in the form of a few seconds of silence delimited speech with known orthography. Using this data a larger system with 4545 tied states and 8 component mixture distributions per state was trained. This system had substantially better performance giving 51.6% word error on the development test data with the bigram language model and 49.5% with the trigram.

6. CONCLUSION

In this paper we have described how the use of word lattices has been incorporated into the HTK large vocabulary system. These have been used to substantially reduce the ammount of time required for optimising current and evaluating new systems. Techniques for evaluating the *quality* of these lattices have been developed in conjunction with an algorithm for pruning them to reduce their size without affecting their accuracy. These techniques have also been used on a corpus with substantially different characteristics and have enabled the construction and evaluation of a recogniser with state-of-the-art performance.

ACKNOWLEDGMENTS

This work is in part supported by an EPSRC/MOD research grant, number GR/J10204. J. Odell is funded by an EPSRC studentship.

REFERENCES

[1] AUBERT X., DUGAST C., NEY H. & STEINBISS V. (1994). 'Large Vocabulary Continuous Speech Recognition Of Wall Street Journal Data.' *Proc. ICASSP'94*, Vol. 2, Adelaide.

[2] NEY H., HAEB-UMBACH R, TRAN B-H. & OERDER M. (1992). 'Improvements in Beam Search for 10000-Word Continuous Speech Recognition.' *Proc. ICASSP'92*, Vol I, San Francisco.

[3] ODELL J.J., VALTCHEV V., WOODLAND P.C. & YOUNG S.J. (1994). 'A One Pass Decoder Design For Large Vocabulary Recognition.' *Proc. ARPA Human Language Technology Workshop, March 1994*. Morgan Kaufmann.

[4] WOODLAND P.C., ODELL J.J., VALTCHEV V. & YOUNG S.J. (1994). 'Large Vocabulary Continuous Speech Recognition Using HTK.' *Proc. ICASSP'94*, Vol. 2, pp. 125-128, Adelaide.

[5] YOUNG S.J., ODELL J.J. & WOODLAND P.C. (1994). 'Tree-Based State Tying for High Accuracy Acoustic Modelling.' *Proc. ARPA Human Language Technology Workshop, March 1994*. Morgan Kaufmann.

[6] YOUNG S.J., WOODLAND P.C. & BYRNE W.J. (1993). 'HTK Version 1.5: User, Reference & Programmer Manuals.' *Cambridge University Engineering Department & Entropic Research Laboratories Inc.*, September 1993.