The CUHTK-Entropic 10xRT Broadcast News Transcription System

J.J. Odell[†], *P.C. Woodland*[†‡] *& T. Hain*[‡]

[†]Entropic Ltd., Compass House, 80-82 Newmarket Road, Cambridge, CB5 8DZ, UK.
Email: {jo,pcw}@entropic.co.uk
[‡]Cambridge University Engineering Department,Trumpington Street, Cambridge CB2 1PZ, UK.
Email: {pcw,th223}@eng.cam.ac.uk

ABSTRACT

This paper describes the development of the CUHTK-Entropic 10xRT Broadcast News Transcription System. Previous HTK broadcast news transcription systems have focused on maximising accuracy with few constraints on compute power available. In order to develop a system running in under 10 times real time on a single CPU, detailed investigation and optimisation of the system architecture and mode of operation was required. This paper outlines those developments and discusses the way in which operation under 10xRT was ensured despite variability of the data to be recognised. On the 1998 test the system produced an average word error rate of 16.1% running in 9.5xRT.

1. Introduction

As the state-of-the-art in broadcast news transcription continues to improve, interest is growing in the range of applications for the transcription of such audio sources. However, most research on such systems is heavily focused on increasing accuracy and there has been relatively few investigations into the trade-offs between recognition accuracy and speed.

With state-of-the-art offline transcriptions systems becoming steadily more complex and elaborate there is considerable work needed to achieve high accuracy at a reasonable computational cost. To spur efforts in this direction the DARPA/NIST 1998 Broadcast News (Hub4) Evaluation included a test investigating the performance of systems running in under 10xRT on a single processor. This paper describes the development of a less than 10xRT version of the HTK broadcast news transcription system.

2. 1997 HTK Broadcast News System

The HTK Broadcast News Transcription System [9] used in the 1997 DARPA/NIST Broadcast News Evaluation used a multiple pass recognition strategy consisting of:

Segmentation. The continuous stream of data is coded and split into small homogeneous segments [2].

Initial Decoding. An initial recognition pass is performed using a set of gender independent cross word context dependent triphone models together with a trigram language model.

Lattice Generation. Using the previously generated hypothesis the the speaker gender of each segment is found, the segments are clustered and then unsupervised maximum likelihood linear regression (MLLR) [1] is used to adapt gender dependent triphone models for each segment cluster. Using these adapted models and a bigram language model a word lattice is generated for each segment. This word lattice is expanded to include more complex language model constraints including a word 4-gram and a class trigram model [6]. The 1-best transcription from these lattices is used as input to the next stage.

Quinphone Model Decoding. More complex quinphone HMMs are used in an iterative adaptation/lattice decoding procedure constrained by the previously generated lattices to refine the transcription.

ROVER. Transcriptions from the quinphone stage and the triphone lattice generation stage are combined to form the final output using a voting mechanism.

While this system produced an error rate of only 15.8% in the 1997 Hub4 evaluation the system ran in approximately 300xRT on a Sun Ultra 2300. Recently an HTK system was needed to participate in the Spoken Document Retrieval (SDR) track of TREC7. This involved transcribing 100 hours of broadcast news material and this was infeasible with such a computationally expensive system.

To develop a faster version of the system for SDR purposes a simpler architecture using fewer decoding passes was needed and the benefits of each of the various stages was examined. The initial decoding pass allowing adaptation of gender dependent models increased accuracy significantly. However further iterations and the use of quinphone models produced relatively little benefit at considerable computational cost.

Therefore the HTK system designed for the TREC7 SDR evaluation used only the initial decoding and the lattice generation stage of the system and furthermore only a 4-gram language model was applied (the interpolated category trigram

was not used). The further stages using quinphone models were discarded (together with the ROVER system combination phase). These changes reduced the runtime to approximately 50xRT but increased the word error rate from 15.8% to 17.4% [4]. The increase in error rate was mainly due to just using the simpler and computationally more efficient triphone models with additional search errors responsible for under 0.2% of the errors. The HTK SDR system was used as the basis for developing a system to operate in less than 10xRT.

3. System Development

The execution time of the HTK SDR system can be broken down into: segmentation 4xRT; first pass decode 10xRT; clustering and adaptation 1xRT; second pass decode/lattice generation 30xRT; and language model application 3xRT.

Although these times were dominated by decoding, a system operating in under 10xRT requires faster operation of all components. The search for these gains were focused in the following areas.

3.1. Platform choice

The choice of computing platform was left open for the DARPA evaluation and an obvious way decrease runtime is to increase the speed of the computer used. However knowing which machine is best for decoding is not easy. Although standard benchmarks, such as SPECint95, are useful for comparing different systems none of them accurately reflects decoding performance.

Accurate comparisons required testing the decoder on each platform of interest. These tests showed that, although a Sun Ultra 2300 gave good performance for coding data, training models and other similar jobs (where compute requirements were dominated by floating point operations), an Intel Pentium II CPU was significantly faster for decoding. By October 1998, Intel had also released the Pentium II Xeon processor in which the 512K L2 cache operates at full rather than half core speed. Therefore a Dell Precision 610 workstation (with a 450MHz Intel Pentium II Xeon with 512Kb L2 cache) was chosen as the main compute platform. In tests the faster cache led to 6-8% faster decoder operation on a Xeon than on the equivalent Pentium II.

	Precision 610	Intel N440BX	Ultra 2300
CPU	Pentium II Xeon 450MHz 512Kb	Pentium II 450MHz 512Kb	Ultra 300MHz 2Mb
OS	Windows NT	RedHat Linux	Sun Solaris
Compiler	Intel 2.4	gcc 2.7.2	Sun 4.2

Table 1: Platform details

Further tests under different operating systems indicated that compiler efficiency was also an important issue. The Intel C compiler (version 2.4) seemed particularly efficient with the decoder running almost 20% faster than when compiled with gcc v2.7.2. Table 1 shows details of the various platforms.

It should be noted that, although for convenience some of the less time critical processes were run on a Sun Ultra 2300 machine or an Intel PII based machine running Linux, (with the main decoding jobs run on the Dell Precision 610 under Windows NT), the runtimes quoted for the final 10xRT system are wall clock times for a single process.

3.2. Segmentation and Classification

The segmenter developed for the 10 times real-time system is a simplified version of the one used in the 1997 and 1998 Hub4 evaluation for the unconstrained compute system[2]. The simplified segmenter achieves similar accuracy yet runs approximately four times faster than the full system. Like the full system it consists of three components: classification, gender dependent phone recognition and smoothing.

	BNeval98_1	BNeval98_2
Classification	0.065	0.065
Adaptation (2x)	0.134	0.134
Phone recognition	0.308	0.307
Smoothing/clustering	0.264	0.325
Overall	0.905 xRT	0.964 xRT

Table 2: Segmentation runtime for two halves of the 1998 Hub4 Evaluation Set (BNeval98_1 and BNeval98_2)

Speed improvements in the classification stage was achieved by reducing the number of MLLR adaptation iterations to 2 with decoding in each iteration. The phone recogniser was simplified by reducing the number of mixture components per state of each phone to 8. Further speedup was achieved by more efficient probability computation and caching in decoding. Compared to the segmentation used for the unconstrained system, this scheme increased the overall word error rate on the 1997 Hub4 evaluation set, BNeval97, by 0.3%. The per frame classification accuracy for non-speech was decreased by 2% absolute while a further 6 seconds of speech were incorrectly discarded. The performance degradation on the 1998 Hub4 evaluation set, BNeval98, was somewhat higher with a 5% absolute reduction in accuracy for non-speech classification and 10 seconds additional discarded speech. Table 2 shows real-time factors for the 1998 Hub4 evaluation set running on a Sun Ultra 2300. Apart from the smoothing and clustering stage all real-time factors appear to be data set independent.

3.3. Decoding Parameters

Experiments were performed using the TREC7 SDR system as a starting point to measure the effect on accuracy of tighter

pruning in the decoder. Experiments varying the decoding parameters indicated that it was possible to reduce decoding times by another factor of two without introducing a large numbers of search errors but that faster decoding (at 1-3 xRT) resulted in a significant number of search errors. Figure 1 shows the word error rate versus runtime (expressed as xRT runtime normalised for a Dell Precision 610) for these tests. These results used a single set of wideband acoustic models trained on the 1997 training data.

More detailed investigation of the effects of each of the decoding parameters together with efficiency improvements and the use of an optimised set of HMMs allowed the determination of an improved set of decoding parameters which increased decoding speed by a factor of two without increasing the error rate.

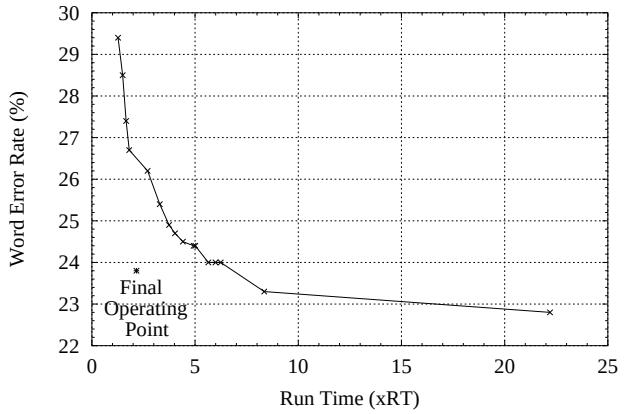

Figure 1: Error rate versus runtime

Also shown in Fig. 1 is the accuracy and speed of this final operating point. Note that although the final operating point used the same dictionary and language model as the other points shown, the optimised HMMs were trained on data from both the 1997 and 1998 training pools which increases accuracy by about 0.9%.

3.4. Variability of decode speed

Another concern when designing a system for guaranteed operation in under 10 times real time is the variability in decoding speed over different segments of speech.

Figure 2 shows a graph of segment cluster decode time (for typical first pass decoder settings) versus duration of the cluster for the first pass decode of BNeval97. This shows that the real time ratio for decoding can vary widely (from 0.71 to 4.81 xRT).

Attempts to limit the CPU time available for processing each frame significantly increased the number of search errors. Consequently, a higher level method of governing the compute requirements was needed. Fortunately, it was found that, despite using different models (together with adaptation), the processing time for the second pass decode was much more closely correlated with the first pass time than the first pass time is with segment duration (an average correlation of 0.95 versus 0.75).

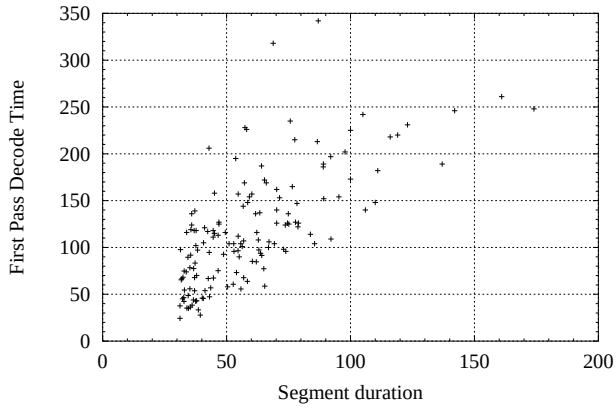

Figure 2: Variation in decode time

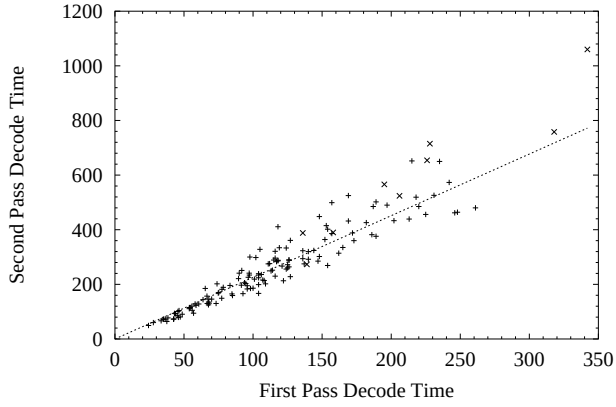

Figure 3: Decode time ratio

Figure 3 shows a graph of second pass decode time versus the first pass decode time. This shows a much more linear and correlated relationship with a maximum ratio between second and first pass times of 3.48 and a minimum of 1.61.

This relationship allows us to accurately predict the CPU requirements of the second pass based on the time taken by the first pass or even to choose the second pass parameters on the basis of predicted second pass time.

The final 10xRT system adopts this latter approach with the decoder operating point for the second pass selected to ensure that the estimated system runtime is under 10xRT. This was accomplished by estimating the second pass/first pass decode

time ratio for a variety of operating points and using a simple decision rule to choose the most appropriate configuration once first pass times were available.

4. Evaluation System Description

The CUHTK-Entropic system used in the 10xRT spoke of the 1998 DARPA Hub4E Evaluation operates in a number of stages. First, the audio is processed by the segmenter which generates three categories of segments: wideband speech, narrowband speech and music. No further processing of music segments takes place and gender assignment of the speech segments is ignored.

For recognition, a 39 dimensional feature vector consisting of 13 MF-PLP cepstral parameters (including c_0) and their first and second differentials is used to represent each frame of data. Cepstral mean normalisation of each segment is applied.

Two sets of cross word triphone context dependent HMMs were produced from the 1997 and 1998 Broadcast news training data supplied by the LDC. The first set of models (HMM1) was used for the initial decoding pass and consisted of 8893 distinct models sharing 4011 tied states each represented by a 16 component Gaussian mixture distribution. State tying was based on decision trees generated by a version of the algorithm described in [11].

The models were initially trained on data coded at the full 8kHz bandwidth. These wideband models were then single pass retrained using the same data with a 125-3750Hz data analysis data to produce a set of narrowband models.

	Word Error Rate (%)		
	BNdev96ue	BNeval97	BNeval98
F0	11.1	12.9	12.1
F1	25.8	19.7	22.2
F2	34.1	27.9	27.5
F3	32.0	32.8	23.9
F4	23.2	25.6	21.8
F5	22.5	25.1	29.4
FX	64.3	43.1	36.5
Overall	26.8	21.4	21.2

Table 3: First pass results

The recogniser used for both the first and second pass is the LVX decoder which forms part of the version 2 release of Entropic's HAPI programming interface [8]. This is a single pass time synchronous decoder incorporating cross word triphones and a trigram language model into a single lattice generating pass. The 4-gram language model is applied to the generated lattice to produce the single most likely hypothesis for each segment.

This hypothesis was then used to determine a gender assignment for each segment as well as estimate a transformation set for each cluster of segments. Table 3 gives the breakdown of results from this first pass into NIST "focus" condition for the BNdev96ue (the 1996 Hub4 unpartioned evaluation development test set), BNeval97 and BNeval98 test sets. The computational requirement for this complete first-pass system is about 3xRT.

Gender determination and subsequent recognition used the gender dependent HMMs (HMM2). These were trained in the same way as the first pass models but consisted of 13428 distinct models sharing 5606 tied states each represented by a 20 components mixture Gaussian. Retraining on narrow band data gave both wide and narrow band versions of this model set for which gender dependent male and female models were produced.

Gender assignment was performed by rescoring the hypothesis produced by the first pass using models representing both genders and selecting the one with the highest likelihood to best represent the segment.

Once the speaker gender had been determined for each segment a top-down clustering of segments for each gender at each bandwidth was performed using the covariance based algorithm described in [3]. This process assigned each segment to a single cluster and an adaptation transformation set was generated for each cluster using the results of the first pass decode as the hypothesis for unsupervised adaptation.

These transform sets were estimated using a computationally efficient approximation to MLLR. Compared to the exact approach described in [5] the accuracy is only slightly reduced (by approximately 0.1% absolute or less than 1% relative) but the computation required to estimate each transformation set is significantly reduced.

The second decoding pass used the transforms estimated for each cluster to adapt the appropriate HMM2 model set and then decode each segments into a lattice of hypotheses.

The toolkit described in [7] was used to interpolate the 4-gram language model with the word category trigram. Each lattice was expanded and augmented to include both the 4-gram language model and the category trigram probabilities. These were then interpolated and the overall language model probability weighted and combined with the acoustic likelihoods (also stored in the lattice) to find the most likely final hypothesis with a modified A* search. Finally an alignment procedure was used to determine the word start and end times to include in the final system output.

Apart from a reduction in the vocabulary from over 65k words to approximately 60k words the language models and dictionaries from the main HTK system [10] were used unchanged.

	Word Error Rate		
	BNdev96ue	BNeval97	BNeval98
F0	8.6	9.4	9.7
F1	22.1	15.8	17.6
F2	26.4	19.7	19.1
F3	25.4	25.4	19.5
F4	17.3	19.3	15.7
F5	19.7	19.4	23.4
FX	59.0	30.1	27.3
Overall	22.1	15.8	16.1

Table 4: Final results for various test sets for the full 10xRT system

The word 4gram model had 5.6 million bigram entries, 9.9 million trigram entries and 7.4 million 4-grams. The category trigram contained 850 thousand bigram and 9.4 million trigram entries for the 1000 categories. The dictionary of pronunciations was derived from the 1993 LIMSI WSJ dictionary, TTS generated pronunciations and hand generated additions/corrections.

The word error rate for the final system is shown in Table 4. The breakdown of the overall computation time for each stage shown in Table 5. Comparing Table 4 with Table 3 the advantage of including two passes can clearly be seen: on BNeval98 the error rate was reduced by 24% while the overall computation was increased by about a factor of three.

Runtime (xRT)	dev96ue	eval97	eval98
Segmentation and coding	1.07	1.10	1.09
First pass decode	2.17	1.94	2.09
Gender determine / transform generation	0.63	0.43	0.44
Second pass decode	4.72	4.83	5.52
Final result generation	0.47	0.45	0.34
Total	9.07	8.75	9.48

Table 5: Runtime breakdown for overall 10xRT system

5. Conclusions

Compared with the full 1998 HTK broadcast news transcription system the 10xRT system uses simpler acoustic models (triphones versus quinphones); no vocal tract length normalisation; no full variance transform and a simplified decoding strategy. Overall the 10xRT system had a word error rate 2.3% absolute (16% relative) higher than the full system [10] (which ran in approximately 300xRT) and the same error rate on the 1997 evaluation data as the full system from a year earlier [9].

Further improvement should be possible (for example by incorporating vocal tract length normalisation) but even so the CUHTK-Entropic system yielded the lowest overall word error rate for systems running in less than 10xRT in the 1998 DARPA broadcast news evaluation.

6. Acknowledgements

Thanks are due to all those who contributed to the main HTK system upon which the 10xRT system was based. Dan Povey helped to speed up the segmenter. This work is in part supported by DARPA and an EPSRC grant on "Multimedia Document Retrieval" reference GR/L49611.

References

1. M J F Gales & P C Woodland. *Mean and Variance Adaptation Within the MLLR Framework.* Computer Speech & Language 1996, Vol. 10, 249-264.
2. T Hain, S E Johnson, A Tuerk, P C Woodland & S J Young. *Segment Generation and Clustering in the HTK Broadcast News Transcription System.* Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop, 133-137.
3. S E Johnson & P C Woodland. *Speaker Clustering Using Direct Maximisation of the MLLR-Adapted Likelihood.* Proc. ICSLP'98, 1775-1779, Sydney.
4. S E Johnson, P Jourlin, G L Moore, K Sparck-Jones & P C Woodland. *The Cambridge University Spoken Document Retrieval System.* Proc. ICASSP'99., 49-52,Phoenix.
5. C J Leggetter & P C Woodland. *Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs.* Computer Speech & Language (1995), Vol. 9, No. 2, 171-186.
6. T R Niesler, E W D Whittaker & P C Woodland. *Comparison of Part-Of-Speech and Automatically Derived Category-Based Language Models for Speech Recognition.* Proc. ICASSP'98, 177-180, Seattle.
7. J J Odell & T R Niesler. *Reference Manual: Lattice and Language Modelling Toolkit V2.0.* Entropic Ltd, 1997.
8. J J Odell, D Wood, D Kershaw, D Ollason, V Valtchev & D Whitehouse. *The HAPI Book: A description of the HTK Application Programming Interface.*, Entropic Ltd, 1998.
9. P C Woodland, T Hain, S E Johnson, T R Niesler, A Tuerk, E.W.D. Whittaker & S J Young. *The 1997 HTK Broadcast News Transcription System.* Proc DARPA Broadcast News Transcription and Understanding Workshop, 41-48, Feb. 1998.
10. P C Woodland, T Hain, G L Moore, T R Niesler, A Tuerk, D Povey & E.W.D. Whittaker. *The 1998 HTK Broadcast News Transcription System: Development and Results.* Proc. DARPA Broadcast News Transcription and Understanding Workshop, March 1999.
11. S J Young, J J Odell & P C Woodland. *Tree-Based State Tying for High Accuracy Acoustic Modelling.* Proc. 1994 ARPA Human Language Technology Workshop, 307-312, Morgan Kaufmann.