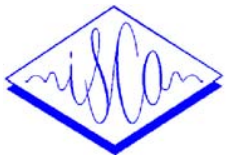

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

TOWARD REALTIME TRANSCRIPTION OF BROADCAST NEWS

Jason Davenport, Long Nguyen, Spyros Matsoukas,
Richard Schwartz, John Makhoul

BBN Technologies, GTE Internetworking
Cambridge, MA 02138 USA
jdavenpo@bbn.com

10.21437/Eurospeech.1999-167x

ABSTRACT

In this paper, we describe our recent work in fast automatic transcription of broadcast news programming from radio and television. Given our state-of-the-art BBN BYBLOS primary system [1] running at 230 times real time (230xRT) we show that eliminating and approximating many computationally expensive components speeds up the system by a factor of more than 20 without significant loss in recognition accuracy. This is accomplished without retraining or changing the base-line system structure. The components of the primary system which are refined include segmentation, adaptation, decoding, cross-word rescoring with adaptation, and system combination.

1. INTRODUCTION

Large vocabulary continuous speech recognition requires a considerable amount of computation. The amount of computation depends to a large degree on the quality of speech, with the computation increasing by a significant factor for more variable speech such as those found in broadcast news. Transcription systems running in research batch mode frequently take 200 to 500 times real time to achieve the highest possible accuracy. While we can always decrease computation by using a straight-forward aggressive pruning strategy with some (if not significant) loss in recognition accuracy, our goal is to balance the speed versus accuracy tradeoff such that the transcription system can run closer to real-time while maintaining similar recognition accuracy as that of a research system.

Since BBN pioneered the public evaluation of the fast Hub-4 broadcast news transcription task in 1997 [2], the ARPA speech research community has (enthusiastically?) participated in this effort through the contrast less-than-ten-time-real-time (10x) Spoke test in November 1998. In this paper, we describe some of the algorithms developed at BBN recently to speed up our base-line 230xRT transcription system to run at around 10xRT to take part in that evaluation. In the next section, we briefly describe our primary time-unlimited system. Following that, we present our 10x Spoke system and compare its speed and accuracy against that of the primary system. Next, we describe in detail some major speedup algorithms used effectively in the 10x Spoke system: Fast Gaussian Computation (FGC), Grammar Spreading, N-Best tree rescoring, and fast and simple adaptation. Then we make our conclusion

2. THE PRIMARY SYSTEM

The overall BBN BYBLOS broadcast news transcription system could be described in these logical steps: (a) Extract speech feature, (b) Segment and classify bandwidth and gender, (c) Cluster the band-specific, gender-specific segments, (d) Decode with Speaker-Independent (SI) models to get transcriptions for adaptation, (e) Adapt models to each cluster, and (f) Decode with Speaker-Adaptively Trained (SAT) models to produce the final answer.

Analysis is performed with a 36-pole LPC model, and Vocal Tract Length Normalization (VTLN). The spectrum mean and variance is normalized over each speaker turn, with speech and non-speech frames normalized separately.

The episode-length waveform is automatically separated into two sets of band-specific long segments using a dual-band, gender-independent, context-independent, phoneme-class decoder. The sets of band-specific segments are then decoded using a dual-gender context-dependent word decoder to separate into two subsets of band-specific gender-specific possibly shorter segments. Speaker change detection and clustering is applied within each band-specific, gender-specific subset to define speaker turns and speaker clusters for unsupervised adaptation.

SI decoding is carried out first by the 2-pass decoder [3] to produce N-Best lists. The first pass of the 2-pass decoder is just a fast-match using a Phonetically Tied-Mixture (PTM) acoustic model and a bigram language model. The second pass, using a within-word State-Clustered Tied-Mixture (SCTM) acoustic model and a trigram language model, is a regular beam search constrained within the most likely words selected by the fast-match pass. The N-best lists are then rescored with a cross-word SCTM acoustic model and trigram language model to produce the top-1 hypothesis to be used in adaptation.

Speaker-Adapted (SA) decoding is a repeat of the SI decoding without the fast-match pass but with adapted acoustic models.

Four decodings are done for each segment -- with 125, 100, and 80 frame/second test set analysis, and a triphone decode with 100 frame/second analysis. System Combination is applied to the outputs of the four systems in order to choose the final answers.

3. THE FAST SYSTEM

The BYBLOS fast transcription system is basically the Primary system reconfigured to exclude the intensive-compute-but-little-gain components and with all speedup options turned on. Table 1 shows a comparison of the primary and fast system through all stages of recognition on the Hub-4 1997 evaluation data set (h4e97). The first column lists the stages of the recognition pipeline. The next 2 columns show the real-time factor for the stages. And the last column lists the accuracy loss (i.e. increase in WER) incurred by speeding up or excluding that stage. Overall, the Primary system runs in 231.2xRT to achieve 14.8% WER. The Fast system performs at 17.5% WER in 9.8xRT, a speedup factor of 23 with a relative loss of 18% in accuracy. All speeds are measured on 450 MHz Pentium II PCs with 512M RAM running the Linux operating system, version 2.32. For comparison, this processor scored 17.2 on the SPECint95 benchmark and 12.9 on SPECfp95.

	Primary xRT	Fast xRT	WER
			14.8
Analysis	0.1	0.1	0.0
VTLN Stretch Estim.	1.5	1.5	0.0
Segmentation	7.9	**1.4**	+0.5
SI 2-Pass Decode	15.4	**4.0**	+0.7
SI xword nbest rescore	5.9	**0.9**	0.0
DSAT nonx decode	22.0	0.0	+0.9
DSAT xword rescore	14.8	**1.9**	+0.2
System Combination	163.6	0.0	+0.4
Total	**231.2**	**9.8**	**17.5**

Table 1. Speed/Accuracy tradeoff for Primary vs. Fast systems measured on h4e97 test set.

We excluded the top two time-consuming stages in the Fast system: *System Combination* and *DSAT nonx decode* (implied by the 0.0xRT factors in column 3). System-combination, also known as ROVER [4], is a hot, trendy, but **impractical**, thing permeating many speech systems recently. It requires a lot of computation, but it is simple to do – just running the same core system with different configurations many times and then voting for the final results at the word level. We decided to trade off a 0.4% WER to save that 163.6xRT. The repeat of the backward pass decode with adapted DSAT model was excluded due to its heavy I/O while loading and reloading speaker-adapted models.

All stages where the real-time factors are shown in bold typeface are sped up by utilizing some or all of the speedup algorithms described in the next section. Furthermore, the N-best lists of the Fast system consist of 100 hypotheses, rather than 300 as in the Primary system, to reduce the time required to rescore them.

4. SPEED-UP ALGORITHMS

Fast Gaussian Computation (FGC) is used in decoding and rescoring to reduce the computation required to evaluate observation probabilities. Grammar spreading is used in the backward pass during the search to achieve robust pruning. N-Best tree rescoring is used to remove redundant computation associated with almost-identical N-Best hypotheses. And the simplified adaptation scheme is used in the adapted cross-word rescoring stage to cut computation further.

4.1 Fast Gaussian Computation

BYBLOS is a fully continuous density system with a mixture of Gaussians modeling the emission at each HMM state. As a result, Gaussian computation is the bottleneck. In the past, we attempted to reduce Gaussian computation by using linear descriptive analysis then choosing the best Gaussians based on a subset of the components of the feature vector. However, this approach did not result in any substantial speedup without incurring significant loss in recognition accuracy. So we abandoned this approach.

In the latest FGC implementation, we use a simpler variation of a decision-tree-based FGC [5]. Conceptually, we would like to partition the whole acoustic feature space into many smaller regions such that for each region, and for any codebook, only a few Gaussians of a codebook can cover that region. These few Gaussians are typically known in the literature as the '*short list*'. We start with the means of all the Gaussians in the system and build a decision tree using binary clustering. Each leaf of this tree can be thought of as representing a unique region of the acoustic feature space. At each leaf we store a short list of the Gaussians from each codebook which are likely in this particular acoustic region. The short lists are made by traversing the tree with training data samples already labeled with codebook and Gaussian ids. If any codebook within a leaf has no samples in the training data, we find the Gaussian that is closest to the mean of the leaf as the sole likely Gaussian for this codebook in this region.

During decoding, for each feature vector we traverse the decision tree to determine its acoustic space region.

Then, for each codebook associated with an HMM state, instead of evaluating all the Gaussians of the codebook, we evaluate only those found in the short list; thus saving computation. The shorter the list is, the more computation can be saved. A schematic visual view of the traversing can be found in Figure 1 below.

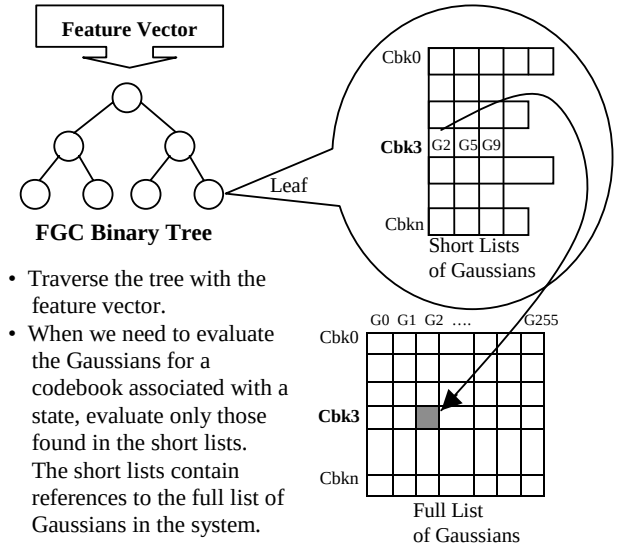

Figure 1. FGC short list lookup during decoding.

In the forward pass of our efficient 2-pass decoder [3], we use a PTM acoustic model with 256 Gaussians per codebook. With FGC, we reduce the average number of Gaussians per codebook to be evaluated during decoding down to 37. And for the backward pass, using a 64-Gaussian SCTM acoustic model, the average number of Gaussians to be evaluated is 23. We show in Table 2 the effect of FGC on the forward and backward pass with a narrow beam on a Hub-4 development test using a 1998 BYBLOS fast system. We can see that the forward pass is sped up by a factor of 3 and the backward pass by a factor of 2.5 with almost no loss in accuracy.

Model	FGC	xRT	WER	# of computed Gaussians / Cbk
Fw-PTM	No	2.3	20.7	256
Fw-PTM	Yes	0.7	20.9	37
Bw-SCTM	No	1.0	20.8	64
Bw-SCTM	Yes	0.4	20.9	23

Table 2. FGC vs. No FGC using a narrow beam.

4.2 Grammar Spreading

During a beam search theories are kept active if their scores are within some factor of the largest path score at that frame (i.e. within the beam). The beam search is clearly not admissible, because a theory that currently scores poorly might later have a better score. The basic algorithm works well if the different theories each get their scores gradually in a time-synchronous manner or the beam is so large that potentially good theories aren't pruned out. In the Primary system we set the beam wide enough so that we don't remove the best scoring global path prematurely. For the Fast system, a wide beam is costly in terms of computation, so we must assure that theory scores are adjusted gradually.

A major cause of abrupt score changes lies with the language model probabilities coming not at every frame, but rather at word transitions. These language model probabilities, which can often be below 10^{-6} for the correct word, occur at different times for each theory. Furthermore, we often exponentiate the language model probabilities by two or more in order to balance them against the acoustic probabilities. Thus, a theory can have its path score decrease in one frame by $10^{-12,}$ which is comparable to the width of the beam that we use in the fast search. When combining these low language model scores with the acoustic score, the theories might be pruned out prematurely.

To prevent prematurely pruning correct theories off a narrow beam, we developed an algorithm to spread the language model probabilities across all the phonemes of a word to eliminate these large score spikes. Figure 2 shows an example of spreading the (backward) bigram P(w2|w1) over the k phonemes of word w2. The transition between w2 and w1 now takes the reduced penalty ratio P(w2|w1)/P(w2), and each transition to each of the k phonemes $P(w2)^{1/k}$ makes up the rest of the penalty.

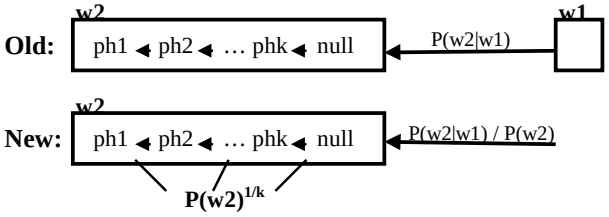

Figure 2. Spreading grammar costs across phones.

We experimented with several types of quantities to spread, such as trigram or bigram weighted average but surprisingly found that spreading the unigram probability worked best.

Spread Grammar	Beam width	XRT	WER
No	Wide	5.2	26.3
No	Medium	1.8	29.7
Yes	Medium	2.0	27.4
Yes	Narrower	1.1	28.6

Table 3. Effect of spreading grammar in backward pass.

In Table 3 we can see the effect of grammar spreading on the backward pass. It shows that we can either reduce computation by a factor of 2 with no loss, or reduce the computation significantly for a small penalty.

4.3 N-Best Tree Rescoring

In the Primary system during the rescoring pass we decode with crossword models each of the 100 or more N-Best hypotheses separately. Typically there are only one or two words that differ in successive hypotheses. For the Fast system we create a tree of these hypotheses for each utterance and score overlapping paths only once. This eliminates the redundancy of scoring identical partial paths of similar hypotheses. The algorithm also allows us to prune more effectively as we are scoring all theory paths in parallel. As described in the grammar spreading subsection, we prune based on a factor of the largest path score, so the beam width has a much larger impact on a tree of multiple hypotheses. We lose no accuracy using the N-Best tree rescorer, and cut the rescoring computation by a factor of 2.

4.4 Simplified Adaptation

In the Primary system we adapt the crossword DSAT models using 2 iterations of MLLR with full matrix transformations to the results of a previous DSAT non-crossword decoding. For the Fast system we adapt the same models but using only 1 iteration of MLLR with 8 block-diagonal transformations to the results of the SI crossword rescoring. Recall that we eliminated the adapted DSAT non-crossword decoding step in the Fast system.

Furthermore, in the Primary system adapted crossword rescoring is done one speaker cluster at a time. The cluster-adapted acoustic model is generated by applying the estimated transformation on the SI model prior to rescoring. This procedure is highly inefficient when there are many speakers in the test set, since a lot of time is spent in I/O and parameter initialization. In the Fast system we avoid this inefficiency by incorporating the adaptation module into the rescorer. We rescore multiple speakers in one step, performing all the necessary I/O and initialization only once. The effect of adapted multiple speaker rescoring is shown in Table 4 where we can see a 68% relative improvement in speed over the primary system with a minimal loss in accuracy.

	XRT	WER
Normal Adaptation	2.82	20.9
Fast Adaptation	0.88	21.1

Table 4. Effect of fast adaptation in rescoring

5. CONCLUSION

We have shown that the BBN BYBLOS transcription system, when utilizing several speedup algorithms described in this paper, can achieve reasonable recognition accuracy at closer to realtime speed on current commodity desktop PCs. Observation probability calculation in a continuous density speech recognition system can be sped up by several factors when using an FGC algorithm to partition the acoustic space into smaller regions covered by short lists of Gaussians. Robust aggressive pruning strategy can be achieved in the beam search when applying the Grammar Spreading algorithm to provide a gradual change in language model probabilities across phonemes of a word (rather than at word boundaries). N-Best hypotheses can be rescored efficiently together by making an N-Best tree, thereby removing redundant computation. And fast adaptation is possible with mathematical approximation and clever software engineering. Nevertheless, 10 times realtime is still far from realtime. We are still looking for more speedup algorithms to make the dream of automatic realtime transcription of broadcast news come true.

ACKNOWLEDGEMENTS

This work was sponsored by the Defense Advanced Research Projects Agency and monitored by the Space and Naval Warfare Systems Command under Contract No. N66001-97-D-8501. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

REFERENCES

1. S. Matsoukas, L. Nguyen, J. Davenport, J. Billa, F. Richardson, M. Siu, D. Liu, R. Schwartz, J. Makhoul, “The 1998 BBN Byblos Primary System Applied to English and Spanish Broadcast News Transcription”, *Proc. of DARPA Broadcast News Transcription and Understanding Workshop*, Herndon, Herndon, VA, Mar. 1999.
2. J. Davenport, L. Nguyen, R. Schwartz, F. Kubala, H. Jin, S. Matsoukas, D. Liu, "Real Time Contrast System Description" *Proc. of DARPA Speech Recognition Workshop*, Chantilly, VA, Feb. 1997.
3. L. Nguyen, R. Schwartz, “Efficient 2-Pass N-Best Decoder” *Proc. of EuroSpeech97*, Rhodes, Greece, Sept. 1997, pp. 167-170.
4. J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)" Draft. *1997 LVCSR DARPA HUB-5E Workshop*, Linthicum, Maryland, May 1997.
5. M. Padmanabhan, E. E. Jan, L. R. Bahl, M. Picheny, "Decision-tree based feature-space quantization for fast Gaussian computation" *Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, Dec. 1997, pp. 325-330.