Efficient Scalable Speech Compression for Scalable Speech Recognition

Naveen Srinivasamurthy, Antonio Ortega, Shrikanth Narayanan

Integrated Media Systems Center, Dept of EE-Systems
University of Southern California
Los Angeles, CA 90089-2654
[snaveen,ortega,shri]@sipi.usc.edu

Abstract

We propose a scalable recognition system for reducing recognition complexity. Scalable recognition can be combined with scalable compression in a distributed speech recognition (DSR) application to reduce both the computational load and the bandwidth requirement at the server. A low complexity preprocessor is used to eliminate the unlikely classes so that the complex recognizer can use the reduced subset of classes to recognize the unknown utterance. It is shown that by using our system it is fairly straightforward to trade-off reductions in complexity for performance degradation. Results of preliminary experiments using the TI-46 word digit database show that the proposed scalable approach can provide a 40% speed up, while operating under 1.05 kbps, compared to the baseline recognition using uncompressed speech.

1. Introduction

In distributed speech recognition (DSR)[1] speech is acquired at the client and the speech recognition is performed at a remote server. A complex recognizer can be implemented on the server enabling low complexity clients to support speech recognition applications. Additionally, in application scenarios where the ambient environment is highly variable, such as those involving mobile devices, there may be much to be gained in terms of running more complex schemes at a remote server for improved recognition. Encoding the feature vectors instead of the raw speech (using a standard speech encoder like FS-10 or MELP) for transmission over a channel reduces the problem of recognition degradation when compression has to be used to reduce bandwidth requirements. Such a system has the added benefit of splitting the computation between the client and the server with the client handling the less complex feature extraction and the server handling the more complex pattern recognition task. We can imagine several clients, each possibly located in different environments accessing the server at the same time. Depending on the (relative) quality of the original captured speech, the task of recognition can be relatively simple or complex. For example, for clients located in relatively noise free environments (“matched”) recognition can be performed with a high degree of accuracy while for clients in a noisy environment recognition accuracy can be improved by interaction between the server and the client (and/or user). In addition to improved recognition quality, DSR is desirable in situations where access to secure data is required. With widespread deployment of DSR we can expect the number of clients accessing the server to grow substantially. As the number of clients increases designing efficient servers becomes more important as it increases the number of clients that can be supported by one server, and this reduces the overall cost. High client density imposes not only computational constraints on the server but it also significantly increases the network traffic at the server. We propose a novel method to tackle both these problems by using scalable recognizers along with scalable encoding schemes. This scheme is shown in Figure 1. The initial low complexity recognizer operating on coarse data can provide a reasonable estimate of the class. This decision can be used to restrict the number of potential classes for the final high complexity recognizer which operates on high resolution data to provide the final recognition result. Often the low complexity recognizer itself can make the final decision, i.e. the number of potential classes is only one, implying that the high complexity recognizer need not be used and the server will not request the client for the enhancement data (effectively reducing the network bandwidth at the server).

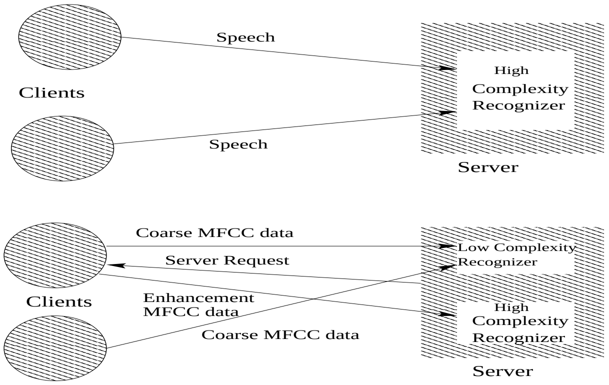

Figure 1: Scalable recognition system. The top figure shows the conventional DSR system. The number of clients that the server can handle is limited by the server computational capabilities and the server bandwidth. The bottom figure shows the proposed scalable system wherein the computational requirements as well as the bandwidth at the server are reduced. In the scalable system the enhancement data is requested by the server only if the low complexity recognizer can not make the recognition decision.

For purposes of scalable recognition, we consider a template-based dynamic time warping (DTW) recognizer and a hidden Markov model (HMM) recognizer for the low and high complexity schemes, respectively. To take full advantage of the recognizer scalability we require a scalable encoder, which will enable reduced bandwidth requirements at the server. Feature vectors are usually derived from speech utterances that have been segmented using overlapping windows. Due to this overlap it is reasonable to expect high correlation between feature vectors corresponding to adjacent frames. This correlation can be exploited by using linear prediction while encoding the feature vectors. Previous work on scalability in predictive coding [2] has addressed the issue of designing a layered coder for video. However this approach requires knowledge (or modeling) of the prediction error *pdf*. We propose a novel scalable encoder motivated by the multiple description encoder design (using a fine and coarse DPCM loop) proposed in [3]. Our proposed scheme can easily be combined with the predictive coding design proposed in [2] if the prediction error *pdf* is modeled. Section 2 provides details of the scalable system design. In Section 3 the experiments and results are presented. Conclusions are presented in Section 4.

2. Scalable encoding and recognition

In this section we present our proposed scalable system, consisting of a scalable encoder providing a coarse base layer and an enhancement layer. The base layer is used by the initial recognizer to provide an initial "guess" of the final class. In general, the complexity of the initial recognizer can be reduced by reducing the complexity of the acoustic models and/or language models and/or pattern recognition schemes. The final recognizer makes use of the initial decision (e.g., hypotheses, word lattices) and the enhancement bits to provide the final recognition decision. We begin by explaining the scalable encoding scheme and the scalable recognition scheme is explained in Subsection 2.2.

2.1. Scalable Encoding

In our previous work [4] we had shown that by using one step linear prediction and uniform quantization of the MFCCs we can achieve good recognition performance. The coding algorithm was able to trade-off recognition performance for reduction in rate. However it had the disadvantage of not supporting incremental refinement in recognition performance, i.e. the recognition achieved at a low rate could not be improved by using refinement bits. Instead to improve the recognition the encoded bitstream corresponding to the finer resolution had to be received. To overcome this drawback we propose a layered scheme wherein the base layer consists of data encoded using a coarse DPCM loop. In the most straightforward approach the enhancement layer can be constructed by encoding the quantization error introduced by the coarse DPCM loop. However it was observed that the bitrate required for the enhancement layer was comparable to that required for an independent fine DPCM loop. An alternate approach would be to maintain both the coarse and fine DPCM loops and use information from the coarse loop to enable better compression of the fine loop prediction error. We propose a novel scheme based on the consistency criteria proposed in [3].

For input sample u_i let e_i and E_i be the prediction errors of the coarse and fine loop DPCMs. Then

$$e_i = u_i - \alpha\hat{u}_{i-1}, \quad E_i = u_i - \alpha\hat{U}_{i-1} \quad \Rightarrow E_i = e_i + \alpha(\hat{u}_{i-1} - \hat{U}_{i-1}) \qquad (1)$$

where $\hat{u}_{i-1}$ and $\hat{U}_{i-1}$ are the reconstructed samples of the coarse and fine loop DPCMs. Let $z_i = (\hat{u}_{i-1} - \hat{U}_{i-1})$, and given that $e_i \in [a_k, b_k]$, the interval R_c in which E_i has to lie can be found as

$$E_i \in R_c = [a_k + \alpha z_i, b_k + \alpha z_i] \qquad (2)$$

Let Q_f be the fine loop DPCM quantizer with N levels, then only the bins of Q_f that intersect R_c are valid choices for the fine DPCM prediction error. This is illustrated in Figure 2 where the region R_c intersects 3 bins (highlighted) of Q_f. If Δ_c and Δ_f are the step sizes used in the coarse and fine DPCM loop quantizer, then the number of valid bins M is at most $\lceil\frac{\Delta_c}{\Delta_f}\rceil + 1$. If $M << N$ then significant savings in bitrate can be achieved.

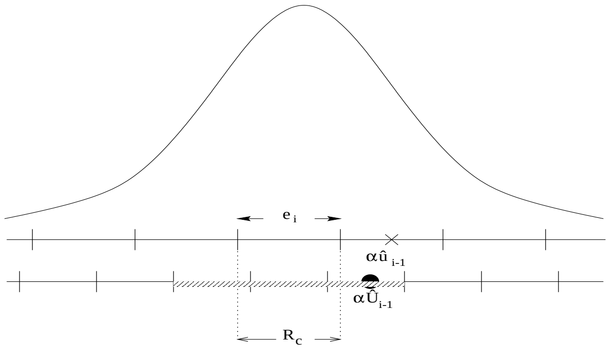

Figure 2: Overlapping shifted quantizers. Using information from the coarse reproduction the number of potential bins for Q_f is reduced. Only the bins of Q_f which overlap the region R_c are valid. The valid bins of Q_f are highlighted.

Let j_i and J_i respectively, be the quantization index of the coarse and fine DPCM loops at time i. Context information from previous coarse and fine reproductions can be used to reduce the entropy of the current fine DPCM prediction error. We have already used $\hat{U}_{i-1}$ in the fine DPCM loop and $\hat{u}_i$ to find the valid bins of Q_f. In addition, we can use the previous reconstructed value of the coarse DPCM loop $\hat{u}_{i-1}$ to bias the probability of occurrences for the different bins J_i of the fine DPCM quantizer. While $\hat{u}_{i-1}$ by itself does not provide explicit information, the difference $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ provides information about bins J_i of the fine DPCM quantizer. Consider the case when $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ is small if $j_i = 0$, then it is highly likely that $J_i = 0$ and if $j_i \neq 0$, then it is highly likely that $J_i \neq 0$. On the contrary when $|\hat{u}_{i-1} - \hat{U}_{i-1}|$ is large, then it is highly likely that $J_i \neq 0$. Using this information we can define two different contexts for J_i

1. $|\hat{u}_{i-1} - \hat{U}_{i-1}| \leq T_q$ and $j_i = 0 \Rightarrow p(J_i = 0) >> p(J_i \neq 0)$
2. $|\hat{u}_{i-1} - \hat{U}_{i-1}| > T_q$ or $j_i \neq 0 \Rightarrow p(J_i \neq 0) >> p(J_i = 0)$

This information can be exploited by using a bitmap to indicate the more probable event in each context, i.e. in context(1) we transmit a "0" if $J_i = 0$ and a "1" otherwise and in context(2) we transmit a "0" if $J_i \neq 0$ and a "1"otherwise. This bitmap, which can be efficiently encoded using run length coding, can be used by the decoder to find the positions where the prediction error is zero. So now the encoder only has to transmit the non-zero coefficients (in addition to the bitmap). Using the information from the coarse loop to find the valid bins and the context information from $\hat{u}_{i-1}$ and $\hat{U}_{i-1}$ we were able to reduce the bitrate for the enhancement layer by about 36%.

2.2. Scalable Recognition

Speech recognition by HMMs has increasing become popular since they provide good recognition results. However, with complex acoustic and language models, they can become a computational bottleneck at the server, when the server is accessed by many clients. The idea here is to consider scalable recognition where we can trade-off complexity versus accuracy. Consider the simple task of isolated digit recognition. Every unknown utterance needs to be scored with 10 models before deciding the best match. One method to reduce the computation would be to speed up the HMM (for example using small model sizes). Another method, which we adopt in this paper, is to build scalable recognizers to restrict the number of the models the recognizer has to operate upon at any time. Usually when an unknown utterance is scored against the different models, only a few of the model scores will be high; this fact can be used to eliminate some of the models from consideration. A low complexity pre-processor can be used to find the N most likely models and the HMM recognizer can be used to choose the best model from these N models. In our example system we choose to use a DTW recognizer as the pre-processor. Since the DTW finds the distance of the unknown utterance from the known templates and the distance is usually minimum between utterances of the same class, we can use a distance threshold to find the N most likely models. An adaptive threshold is used for every utterance based on the lowest distance obtained after template matching. Adaptive threshold in contrast to a fixed threshold has the desirable feature of selecting more models when the distance (likelihood) between the best and other models is close, and selecting a few (sometimes only one) models when the distance between the best and other models is far. The procedure for recognizing an unknown utterance using the above system is

Algorithm 1 *(Scalable Recognizer : System A)*
Step 1 : *Find the distance $D(k)$ between the unknown utterance and the L templates using DTW.*
Step 2 : *Select the models with distance $D(k)/D(0) < T$.*
Step 3 : *Use HMM to find the best model among the chosen models in* **Step 2***.*

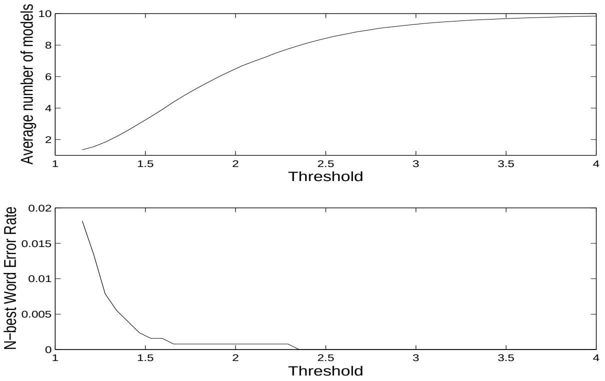

Figure 3: Observe that the average number of models increases with threshold and the probability of word error monotonically decreases with threshold. At threshold of 2.4 the probability of word error becomes zero but the average number of models is reduced from 10 to 8, i.e, a 20% reduction in HMM running time with no difference in recognition performance. In general it is not required to use a threshold as high as 2.4, a smaller value between 1.2 to 1.8 will suffice, because utterances very difficult to distinguish by DTW are most likely to be in error for the HMM also.

Figure 3 shows the average number of models retained after the initial DTW stage and the probability of word error in the N-best list of the DTW as a function of the threshold T. The average number of models monotonically increases with threshold and the probability of word error monotonically decreases with threshold. This fact can be used to trade-off between complexity and recognition-performance. A low threshold would imply that the WER would be high but the complexity would be low and vice-versa. A simple observation is that with a threshold of 2.4 we get no word error however the average number of models is reduced from 10 to 8 by the use of the initial DTW stage.

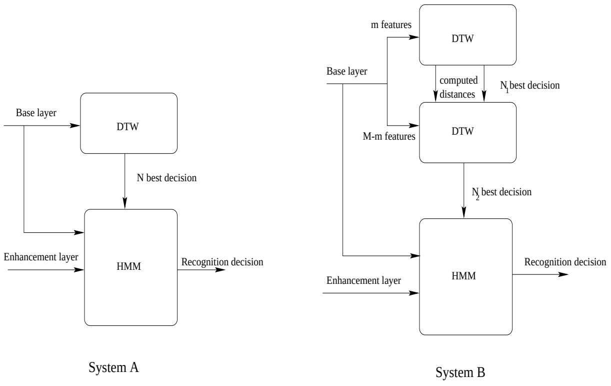

Figure 4: The different scalable recognizer schemes used. In System A we use a DTW as the initial recognizer and a HMM as the final level recognizer. In System B we use two levels of DTW as the initial recognizer and a HMM as the final recognizer.

To further reduce the complexity we can use 2 stages of DTW before using the HMM recognizer. The initial DTW stage uses the MFC coefficients deemed more important to generate an N_1 best list. The second DTW operates on these N_1 models and the distance is refined with the remaining MFCCs to generate an N_2 best list. The HMM makes its decision from these N_2 models. This procedure is summarized below.

Algorithm 2 *(Scalable Recognizer : System B)*
Step 1 : *Find the distance $D(k)$ between the unknown utterance and the L templates using m of the M MFCCs.*
Step 2 : *Select the models with distance $D(k)/D(0) < T_1$.*
Step 3 : *Refine the distances $D(k)$ for the models chosen in* **Step 2** *using the remaining $M - m$ MFCCs.*
Step 4 : *Select the models with distance $D(k)/D(0) < T_2$.*
Step 5 : *Use HMM to find the best model among the chosen models in* **Step 4***.*

Importance of the MFCCs can be determined by dropping one MFCC at a time and finding the effect on N best recognition by DTW. The MFCC that introduces the most error in recognition is declared as the most important and so on. From our experiments the coefficients were ordered from most important to least important as [2,3,0,1,6,7,4,8,5,10,11,9]. Note that for both the algorithms the number of models retained at each intermediate step can be variable depending on the unknown utterance. Also if at any intermediate step the number of models retained is only one, then the subsequent recognizers need not be used and the unknown utterance is classified as the digit corresponding to the retained model. As before, thresholds T_1 and T_2 can be varied to trade-off between complexity and recognition-performance. Since the initial stage(s) is(are) used primarily to speed up the recognition operation, we can use the base layer for the decision process. During refinement by the HMM the enhancement layer can be used to enable more accurate representation of the feature vectors. The DTW computation can be reduced by exploiting the fact that the input data has been predicted and quantized. If the prediction error for the entire frame is quantized to zero, we do not need to find the distance of this frame from all the reference frames, instead the distance computed for the previous frame can be repeated without incurring significant degradation. When the proposed scalable system is used the computational load at the server is reduced. In addition this can be advantageous even from the users perspective if often the DTW recognizer is able to make the final decision since the latency from uttering the speech to recognizing it is reduced. Figure 4 shows the recognition system for System A & B. The recognition performance obtained by the above methods is shown in Figure 5 and Figure 6 shows the complexity (in sec) for these methods.

3. Experiments and Results

The TI46-Word digit database was used for evaluating the proposed scalable system. An HMM (HTK 3.0) based recognizer was used as the final recognizer. The speech utterance was segmented using overlapping Hamming window of length 24 ms, with adjacent windows separated by 12 ms. 12 MFCCs derived from each segment of the speech utterance was used as the front-end. A left to right HMM with 5 states and 2 Gaussian mixtures was trained for every digit using unquantized MFCCs from about 800 utterances from 8 male speakers. Test utterances were from the same speakers of the training data, but different utterances. The training and test utterances have silence periods before and after the digit. To model this a five state silence HMM was trained. The word network included the silence HMM before and after every digit HMM. The baseline WER and complexity were determined by recognizing speech with unquantized MFCCs using only the HMM.

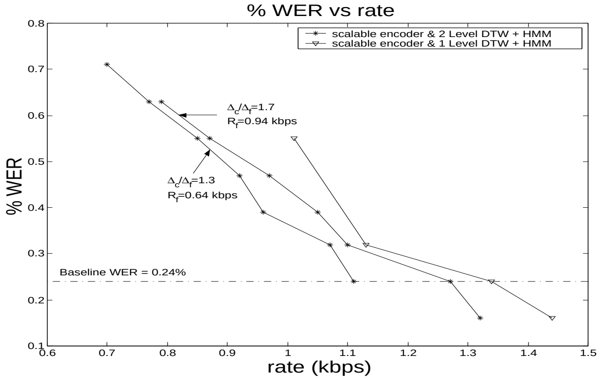

Figure 5: Recognition performance for the scalable recognition schemes. The coarse bitrate is 0.72 kbps. The fine bitrate (R_f) is indicated in the figure. The baseline WER of 0.24% is shown as a dotted line.

Figure 5 shows the WER as a function of average rate for System A & System B (the results for two different Δ_c/Δ_f ratios is shown). We observe that the WER is 0.32% and 0.24% at 1.13 kbps and 1.11 kbps for System A and System B respectively, when compared to the baseline WER of 0.24%. The recognition-rate trade-off is better when Δ_c/Δ_f is smaller, however larger Δ_c/Δ_f provides a superior reconstruction of the enhancement layer enabling lower WER. Figure 6 shows the trade-off in complexity and WER for both the Systems. Using only the HMM we required about 28 sec to recognize 1267 digit utterances (approximately 1400 sec of speech). We reduced the complexity by 21% and 25% by using System A and System B respectively with no degradation in WER. However if we are willing to tolerate more error we can reduce the running time by 53% (i.e. more than halve the running time) by using System B while incurring a WER of 0.63% (163% increase over the baseline performance).

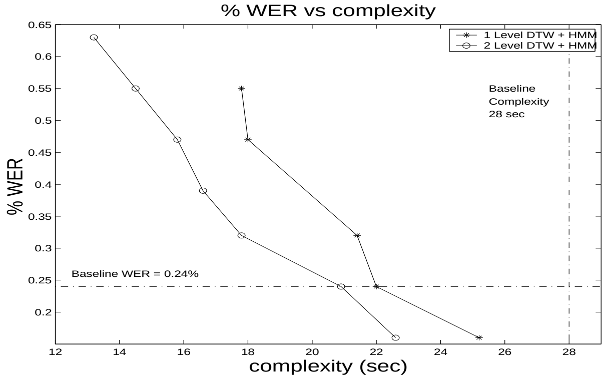

Figure 6: The time required for the scalable recognition schemes. The baseline computational time of 28 sec is shown as a dotted line. Notice the computational scalability wherein reduced complexity can be achieved at the expense of WER.

4. Conclusions

We have proposed the use of scalable recognizers and scalable encoders for distributed speech recognition. Eliminating unlikely candidates using a simple pre-processor before performing the recognition with a more complex HMM enables significant savings in computation. The added advantage of such a system is that it allows a fairly straightforward method to trade-off between complexity and recognition performance. When a scalable encoding scheme is available the scalable recognizers can be used in a DSR application to reduce both the computation and the bandwidth requirements at the server. This would naturally translate into the server being able to support more clients with the same resources. The WER degradation introduced by compression can be reduced by alleviating the “mismatch” between the testing and training phases by the use of model transformations to optimize classification by ensuring that the adapted models are more likely to have produced the observed data [5].

5. References

[1] “Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms,” Tech. Rep. Standard ES 201 108, European Telecommunications Standards Institute (ETSI), April 11 2000.

[2] K. Rose and S. Regunathan, “Towards optimal scalability in predictive video coding,” in *ICIP 98*, pp. 929 –933, 1998.

[3] R. Singh and A. Ortega, “Erasure recovery in predictive coding enviornments using multiple description coding,” in *IEEE Workshop on Multimedia Signal Processing*, 1999.

[4] N. Srinivasamurthy, A. Ortega, Q. Zhu, and A. Alwan, “Towards efficient and scalable speech compression schemes for robust speech recognition applications,” in *ICME 2000*, July 2000.

[5] N. Srinivasamurthy, S. Narayanan, and A. Ortega, “Use of model transformations for distributed speech recognition,” to appear in *ISCA Workshop on Adaptation Methods for Speech recognition*, (Sophia Antipolis), August 2001.