Compression for Speech Recognition and Music Classification

1. Research Team

Project Leader: Prof. Antonio Ortega, *Electrical Engineering*

Other Faculty: Prof. Shrikanth Narayanan, *Electrical Engineering*

Graduate Students: Naveen Srinivasamurthy

Undergraduate Students: Sam Bagwell, Rahul Maini, Merrick Mosst

2. Statement of Project Goals

One of the goals of this project is to develop methods for compressing speech signals for a distributed speech recognition task. The objective of current speech compression techniques is to minimize perceptual distortion. In this project, however, we investigate efficient compression techniques that achieve low bit rate transmission, while incurring a minimal degradation of automatic speech recognition accuracy (as compared to the performance with uncompressed data). Intended applications of this project will be in cases where speech acquisition is done using low power, and possibly mobile, devices while the more complex speech recognition task is performed at a remote server [3]. This framework can be used either on the Internet or in wireless networks.

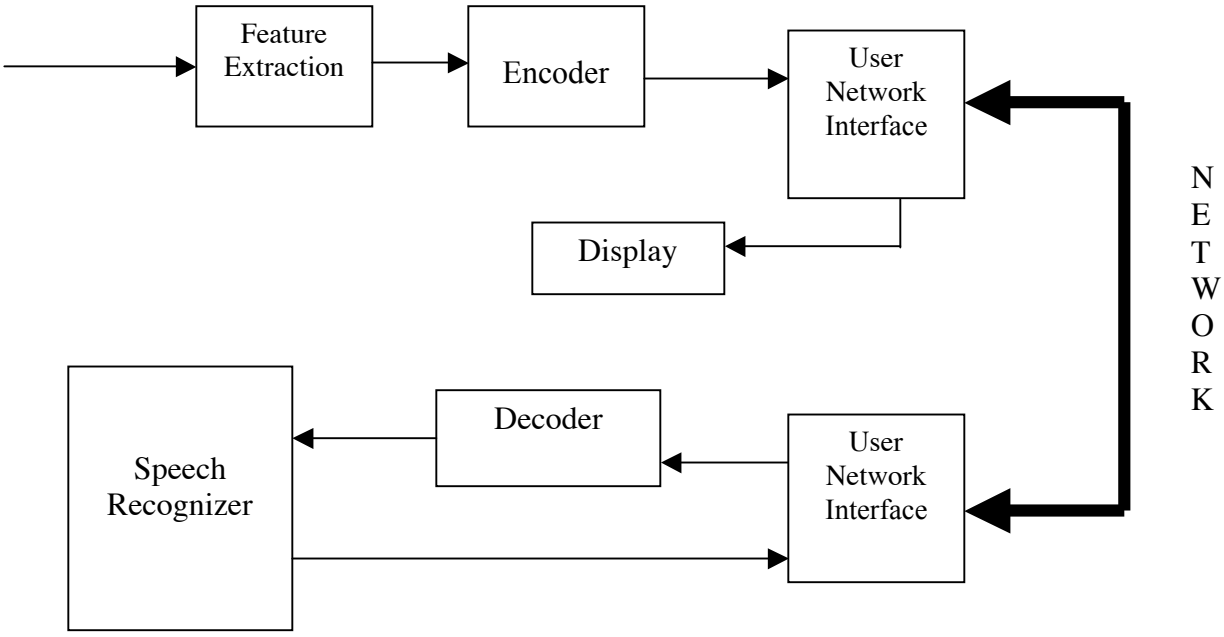
In addition we are studying approaches to measure the similarity between music files. We are working on improving some previously proposed techniques [16,17], and our following step will be to develop techniques so that the features used in comparing music files can themselves be compressed. One key element of this project is to try to establish links between low level features and perceptual similarity as perceived by humans. For this purpose we have developed web-based tests to determine the relative importance of the low level features.

3. Project Role in Support of IMSC Strategic Plan

This work is being developed with a close collaboration between a speech analysis-recognition group and a compression group. We foresee two major applications for this work. First in very low rate MIE scenarios (wireless links) we provide functionality for a mobile user to interact with a remote database through a speech recognition interface. Second, this work, along with its extensions to image/video is useful for multimedia databases, where the data is stored in compressed format and visual features such as color, texture, shape have to be extracted and used as indexing keys. The key difference between prior efforts and our ongoing research is that we consider distributed systems where the end-user is accessing a remote database or system and thus only limited bandwidth is available, and the impact of the transmission latency must also be taken into account. Refer to [12] for detailed information.

4. Discussion of Methodology Used

The scenario we consider is a distributed speech recognizer system as shown in Figure 1. The speech is acquired at the client, which can be a mobile device with limited computational capabilities. The features required for recognition are extracted from the acquired speech at the client. These features are quantized and transmitted to a remote server hosting the speech recognizer. This architecture allows low complexity (possibly mobile) devices to support speech recognition applications. One of the main challenges in a DSR system is to develop a speech-coding algorithm, which minimizes recognition degradation rather then minimizing perceptual distortion. In our recognition system we have used Hidden Markov Models [1] as the speech recognizer. 12 Mel-Frequency Cepstral Coefficients (MFCC) [2] are extracted for every acoustic frame.



5. Short Description of Achievements in Previous Years

We use a one-step prediction of the MFCC vector, wherein the current vector is predicted from the previous vector. The prediction error is quantized using either an ECSQ or USQ. For the ECSQ case different scalar quantizers are designed for every coefficient in the acoustic frame. Importance of each of the coefficients, in the acoustic frame, towards recognition performance is found. Based on the importance of the coefficients we can prune the acoustic frame by dropping some of the coefficients at the encoder, in order to achieve lower bit rates. For best performance we retain a different number of coefficients in every acoustic frame based on the importance of each coefficient in the acoustic frame. Dropping coefficients is made transparent to the speech recognizer because the decoder replaces by zeros all those coefficients that were dropped before inputting the frame into the speech recognizer. A block diagram of a Client-Server model to implement the proposed idea is shown in Figure 1.

Instead of explicitly pruning the coefficients better trade-off between the bit rate and recognition is possible by implicit pruning. Here the assumption is that larger prediction errors are more important. By using USQ with a dead zone we can easily set small prediction errors to zero. Scalability is achieved by changing the quantization step size: coarser step size results in lower rate and vice versa. Moreover, whenever a prediction error is quantized to zero we use the predicted value for the coefficient (rather than set the coefficient to zero), which affects less the recognition performance than pruning the coefficient altogether.

Entropy encoding is used to encode the quantization indices. This is combined with a run length encoder, which is used to encode the bitmap, which indicates to the decoder the position of the non-zero coefficients in each frame. Since we use a bitmap only the non-zero coefficients need to be encoded and this enables us to achieve even lower bit rates.

The algorithms we are developing [10] are scalable, that is, we can choose to lower the bit rate (for example in cases when the bandwidth is lower) by dropping more coefficients at the encoder. For example, our current algorithm when tested on a digit database is capable of operating between 0.77 kbps and 2 kbps, with recognition performances for these bit rates being 98% and 99% respectively. The corresponding recognition performance for uncompressed acoustic features for the digit database is 99.8%. Thus, the degradation in recognition performance by using our method is minimal. For an alphabet database we achieved recognition of 80% and 82 % for bit rates of 0.8 kbps and 2 kbps respectively. The result obtained with uncompressed acoustic features was 82.66 %.

While the method proposed in [6] has similar recognition performance for the digit database, it suffers from the drawbacks of higher bit rate, higher encoding complexity and no scalability. The bit rates for the method proposed in [5] ranged from 2.6 kbps to 10.4 kbps, with recognition performance being 91% and 93.5% respectively.

We achieve the same or better recognition performance than that achieved by the methods proposed in [5.6] using bit rates lower than 1 kbps. Also our encoding algorithms are scalable, allowing a bit rate and recognition performance trade-off, and can be combined with unequal error protection or prioritization to allow graceful degradation of performance in the presence of channel errors. Performance results are shown in Figures 2 and 3. Table 1 shows the CPU time required to recognize and encode an utterance from the digit database.

Speech Recognition	ECSQ	USQ
0.156 s	0.067s	0.047s

Table 1
Cpu time on a Sun workstation to recognize an utterance from the digit database and time required to encode it

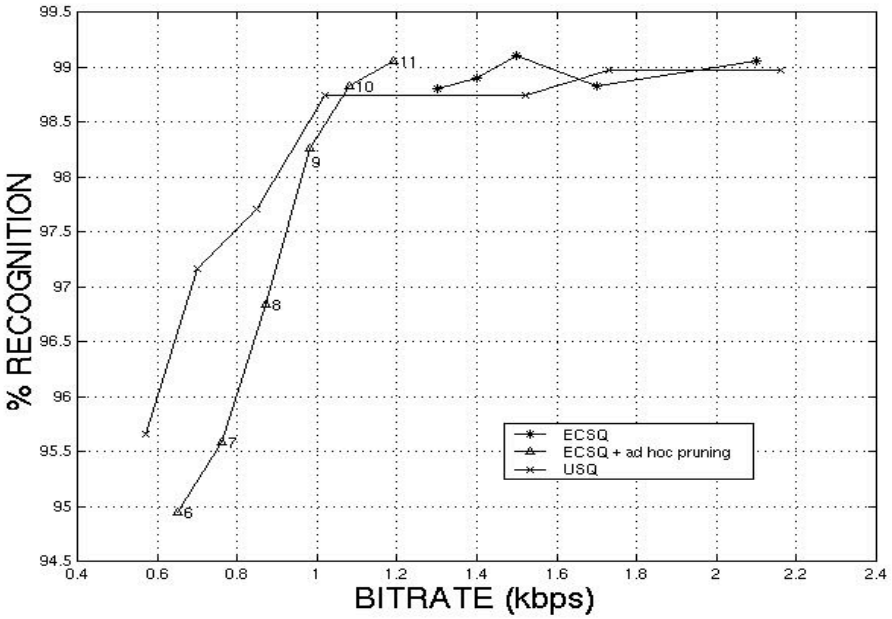


Figure 2. Recognition performance of the different encoders for the digit database

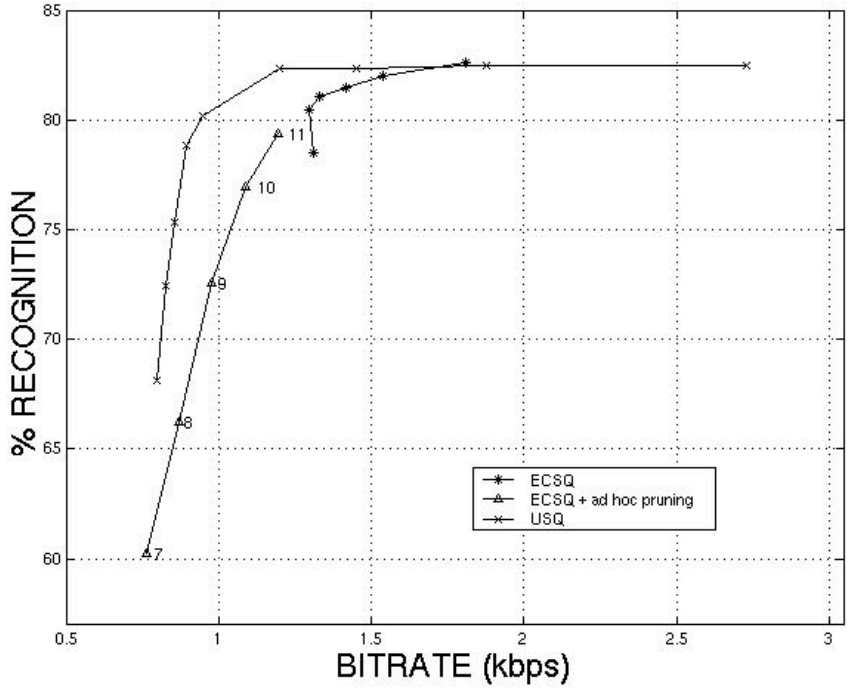


Figure 3. Recognition performance of the different encoders for the alphabet database

Scalable distributed speech recognition

A completely scalable distributed speech recognition (DSR) system combines the scalable encoder with a scalable speech recognizer. Such a system enables reduction of both computational load and bandwidth at the server. A low complexity pre-processor is used at the server to eliminate the unlikely classes so that the complex recognizer can use the reduced subset of classes to recognize the unknown utterance. To reduce the bandwidth requirements at the client, the pre-processor operates on the base layer of the compressed bitstream. When the pre-processor can not make the recognition decision, the server requests for additional layer(s) from the client and the final recognition stage operates on the refined data. A block diagram of this system is shown in Figure 4. With this scalable system it is fairly straightforward to trade-off between complexity, bandwidth and recognition performance. As a proof of concept we implemented a scalable system [13], which used a template-based dynamic time warping (DTW) recognizer and a hidden Markov model (HMM) recognizer for the low and high complexity schemes, respectively. A novel layered DPCM encoder based on the consistency criteria proposed in [14], is used at the client.

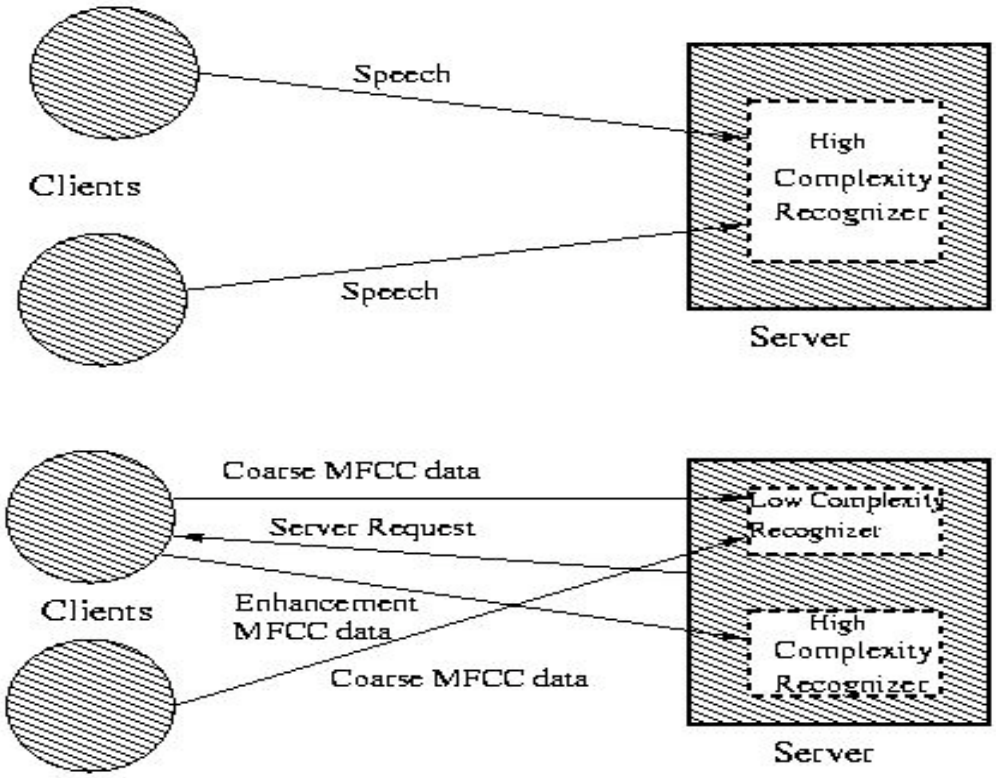


Figure 4. Scalable distributed speech recognition system

The low complexity recognizer operates on the coarse MFCC data to make the recognition decision. The high complexity recognizer uses this recognition decision and the enhancement MFCC data to make the final recognition decision.

The most important performance metrics in a DSR system are:

- User delay
- Client bandwidth
- Server bandwidth
- Server complexity

The operation of the scalable system can be changed depending on which of the above performance metrics is to be optimized. In what follows we describe how our techniques can be used under three different scenarios, with different optimization criteria.

User delay minimization

When the most important constraint is the time taken between the user speaking and the result of recognition, we can generate both the base and enhancement layers immediately and transmit both to the server. The server uses the base layer with the DTW and if only one model is present in the N-best list, the result is sent back to the client; if the N-best list contains more than one model, the enhancement layer (which is already available at the server) is used by the HMM to get the final recognition result. By this method we can ensure that the delay experienced by the user is minimized, while also keeping low the server complexity. However, in this scenario, the client and server bandwidth requirements will be increased.

Client and Sever bandwidth minimization

In bandwidth-constrained situations, initially only the base layer is transmitted to the server. After the DTW stage, if required the enhancement layer is requested from the client. As can be seen from this procedure, both client and server bandwidths can be low, and the server complexity can also be kept low. However the absolute delay experienced by the user can be high (for cases where the DTW is not able to make the final decision).

Server complexity minimization

Irrespective of all other constraints, we can always ensure that the complexity at the server is reduced, as mentioned in the above two cases. However when user delay is a constraint, the memory requirements at the server will be increased since the enhancement layer will have to be stored for future use.

Figure 5 shows the effect on recognition performance as the bit rate is changed it is reduced. With 2 levels of DTW followed by an HMM speech recognizer we were able to achieve the same word error rate (WER), 0.24 %, as with unquantized MFFCs at a bit rate of 1.1 kbps. If we are willing to tolerate higher WER we can further reduce the bit rate (0.63 % WER at 0.77 kbps). Figure 6 shows the trade-off between recognition performance and complexity. Using only an HMM requires about 28 sec to recognize 1267 utterances from TI46 database (about 1400 sec of speech). With the scalable system this can be reduced to 20.5 sec with no increase in WER. Again if we are willing to tolerate higher WER the time required can be reduced to 14 sec (a 50% reduction in server complexity).

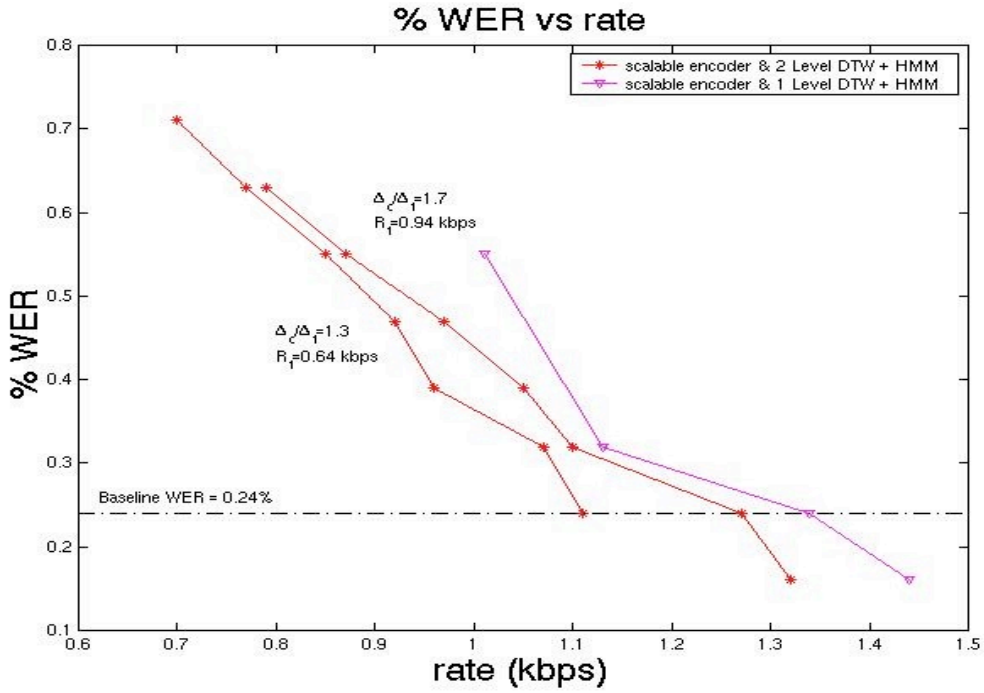


Figure 5. Recognition performance of the scalable DSR in a bandwidth constrained scenario

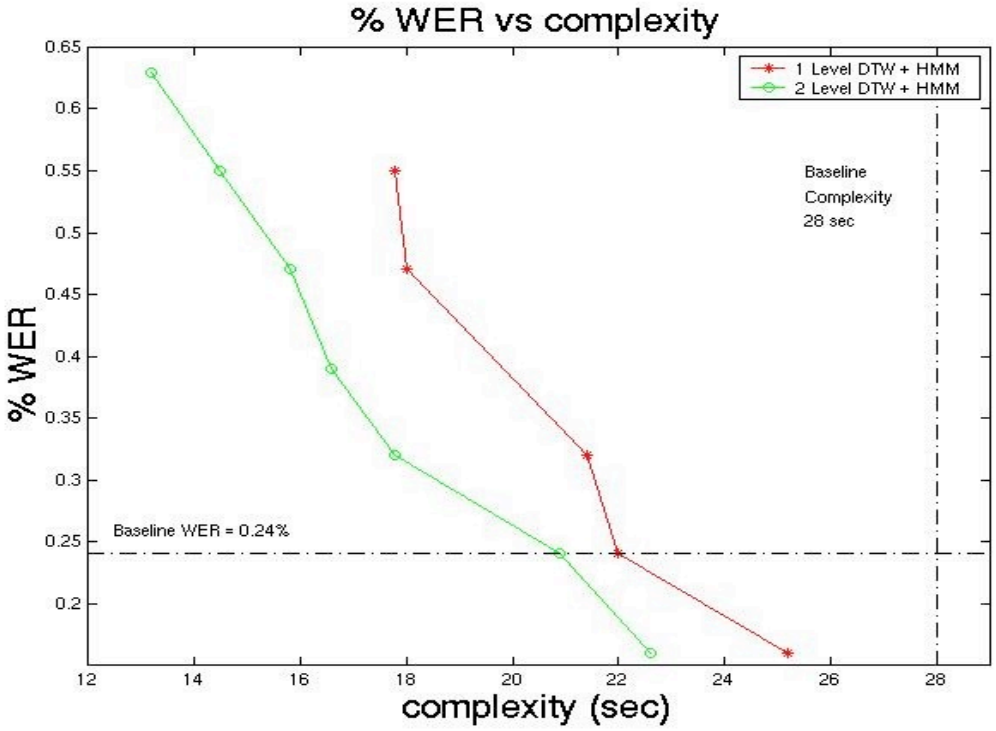


Figure 6. Recognition performance of the scalable DSR in a complexity constrained scenario

Model adaptation for distributed speech recognition

It is well known that for a given task, speech models trained on task-dependent data usually outperform models trained on task-independent data. One of the major problems for robust speech recognition is the mismatch between the training and testing conditions. Speech recognition performance, with speech models trained on clean data, significantly degrades when the test utterances are noisy (channel noise, ambient environment noise). Similarly the performance is also degraded due to long term and short term speaker variations. Also speaker dependent models are known to usually outperform speaker independent models. With wider use of speech recognition applications especially in mobile devices we have an additional source for mismatch: speech encoding. The distortion introduced by speech encoders can also be thought of as a mismatch between the training and testing conditions. It is relatively easy to remove this mismatch by the use of a family of models each trained with data from different encoding schemes, and choose the one that best matches the unknown test data. However, this scheme is not attractive since it might not be possible to have models trained for all the different compression schemes, because the choice of the compression scheme used by the client depends on the computational resources/load at the client and the quality of service (QoS) it wishes to provide the user. Scalable encoders, which could be combined with scalable recognition schemes, further complicate the creation of pre-defined models. Depending on the optimization criteria used for compression (classification vs. human perception), more variability in the compression schemes used by the different clients can be expected.

This mismatch introduced by the choice of different speech compression schemes can be solved in similar manner as other mismatches [15]. The models at the server can be trained using clean speech (or a particular compression scheme) and alleviate the mismatch between testing and training phases by the use of model transformation/adaptation to optimize classification by ensuring that the transformed/adapted models are more likely to have produced the observed data. Note that simple signal processing techniques are not likely to be helpful as the distortion introduced by compression is not invertible, however adaptation schemes which operate on the models rather than the input speech are more likely to be able to reduce the mismatch. The two popular adaptation techniques used previously are MLLR and MAP. Table 2 shows the recognition performance for the TIDIGITS connected digits database for different encoding schemes. MFCC-LR and MFCC-HR are the linear prediction based quantization schemes described before operating at 1.22 kbps and 2.07 kbps. We can observe that with adaptation we are able to significantly reduce the WER for the different encoding schemes. For the MFCC-HR encoder with MLLR adaptation the recognition performance is almost the same as that achieved with clean speech.

Compression	Clean Models	MLLR Adaptation	MAP Adaptation
Clean Speech	1.88	1.57	0.67
MELP	3.14	2.32	1.19
GSM	2.50	1.73	0.91
MFCC-HR	4.81	2.24	3.34
MFCC-LR	2.10	1.60	0.91

Table 2
Model adaptation for different encoding schemes

5a. Detail of Accomplishments During the Past Year

Scalable distributed spoken names recognition task

The isolated digits task is a low perplexity task; here we consider scalable recognition for a high perplexity task. Consider a two stage spoken name recognizer with dictionary lookup (Figure [7]). Such a two stage approach has been used for spelled name retrieval [18], information retrieval [19], complexity reduction of a phone based continuously speech recognition system [20] and spoken name recognition [21]. In the first stage a low complexity bigram phone loop is used to identify the N-best phone sequence corresponding to the input utterance. The next step involves a string match, where each of the N-best phone sequences is compared to the entries in a dictionary. The utterances corresponding to phone sequences, which have a distance less than a given threshold (the threshold is usually chosen as a function of the number of phones in the recognized phone sequence) from the recognized N-best phone sequence are selected to generate a lattice. The final stage involves rescoring this generated lattice using more complex acoustic (triphone) models.

Consider a spoken names recognition task [22,23], which among other applications is used in network-based applications e.g., directory assistance, and caller identification. In these applications the list of names tends to be quite large, in the order of hundreds of thousands. Variability in pronunciation further increases the perplexity. The traditional approach to name recognition has been to use a finite state grammar (FSG), where all the names (with all possible pronunciation variants) are alternate paths for recognition. For a name utterance the recognizer evaluates all possible paths and selects the name corresponding to the most likely path. As the names list grows it is evident that the computational complexity increases and the recognition accuracy will drop. An alternative approach with reduced computational complexity is to adopt a two-stage recognizer with dictionary lookup. Figure [7] illustrates this approach. The accuracy obtained by the two stage recognizer for spoken names task is comparable to the conventional single stage FSG based approach (as shown in [21]) but results in significant savings in complexity, since the lattice only consists of a subset of the entire names list. The dictionary used for lookup is a names dictionary, which consists of all possible names along with their pronunciations. In our experiments we used the Levenshtein (or edit) distance during dictionary lookup to compute the string match distance between phone sequences. The Levenshtein distance between phone sequences p_1 and p_2, LD (p_1, p_2), is the minimum cost associated in transforming p_1 into p_2 by deletions, insertions and substitutions.

This two stage names recognition procedure is summarized below.
Scalable Spoken Names Recognition
Step 1: Identify the N-best phone sequences p_r^n for the name utterance using a bigram phone loop, for n=0,1,..., N-1.
Step 2: Find T_n corresponding to p_r^n from Table [3], for n=0,1,…, N-1.
Step 3a: Initialize i = 0
Step 3b: For name i in the names dictionary find the corresponding phone sequence p_i.
Step 3c: If LD $(p_r^n, p_i) < T_n$, for any n=0,1,…, N-1, add name i to the names lattice.
Step 3d: If there are more names in the dictionary set i=i+1 and go to Step 3b else go to **Step 4**.
Step 4: Rescore the names lattice using context-dependent models to get the final result.

When the spoken names recognition system is used in a DSR system [25] with a variable-rate encoder, both recognition stages, i.e., the phone loop and the lattice recognizer operate on the same compressed data. The phone loop is a bigram CI phone loop. The resulting lattice after dictionary lookup can be refined using either CI or CD models. The results obtained for these two cases for different bit rates is shown in Figures [8] and [9]. We observe that in both cases there is small degradation in performance when the bit rate is greater than 2500 b/s. However it is clear that with the proposed encoder we can trade bit rate for recognition performance. The average number of names in the lattice when compressed data was used was approximately 1140, which is almost the same as when uncompressed data was used, i.e., compression did not increase the lattice size.

Table 4 compares the degradation due to compression with the proposed encoder and *Aurora* encoder [24]. Notice that the proposed encoder provides both a lower rate and has a lower WER degradation than *Aurora*.

Length of phone sequence	Threshold
less than 4	3
4 or 5	4
greater then 5	5

Table 3: Thresholds used during dictionary lookup

Encoding technique	CI	CD	Rate (b/s)
Aurora	0.86	0.77	4400
Proposed variable rate encoder	0.33	0.13	4050
	0.36	0.47	3600

Table 4: Absolute percentage increase in WER for the proposed encoder and *Aurora*. Observe that even when the proposed encoder operates at 3600 b/s it is superior to *Aurora*

Scalable DSR encoder
When the proposed scalable DSR encoder is used at the client, a base layer and an enhancement layer are transmitted to the server for every name utterance. Now the bigram CI phone loop uses the base layer to generate the N-best phone sequence. This is used by the dictionary lookup to build the list of names for the lattice recognizer. The lattice recognizer rescores the names list using the enhancement layer data to get the final recognized name result. Note that the phone

recognizer and the dictionary lookup need not wait for the enhancement layer data to be received.

The recognition results obtained with the above procedure for the names task are shown in Figures [10] and [11]. Observe that when the base layer rate is 2580 b/s and the enhancement layer is 2000 b/s the recognition result obtained with CD models is the same as that obtained with a variable-rate encoder at 4040 b/s (Figure [9]). Because the base layer rate is only 64% of the rate used by the variable-rate encoder the first stage recognizer (phone recognizer and dictionary lookup) completes more quickly which ensures that the second stage (lattice recognizer) can begin earlier and complete much faster in systems employing a scalable coder while achieving the same recognition performance (however there is a 13% increase in rate by adopting a scalable scheme).

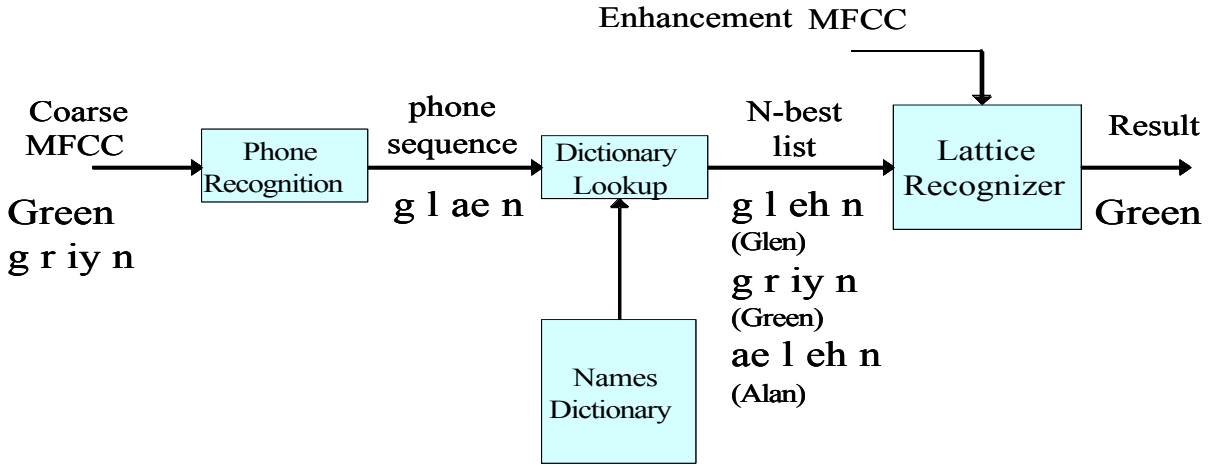


Figure 7: Two stage names recognition approach using dictionary lookup.

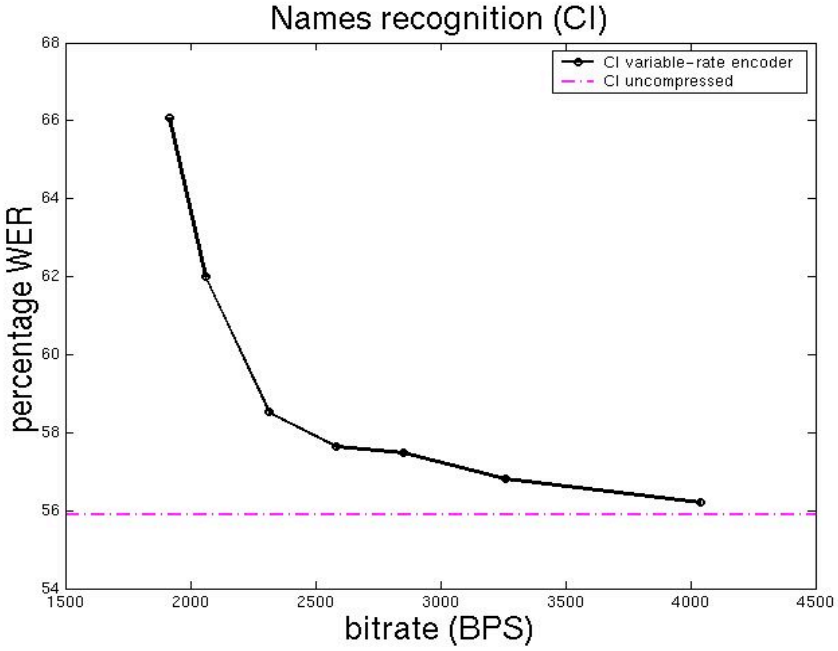


Figure 8: Names recognition results when CI models are used in the lattice recognizer.

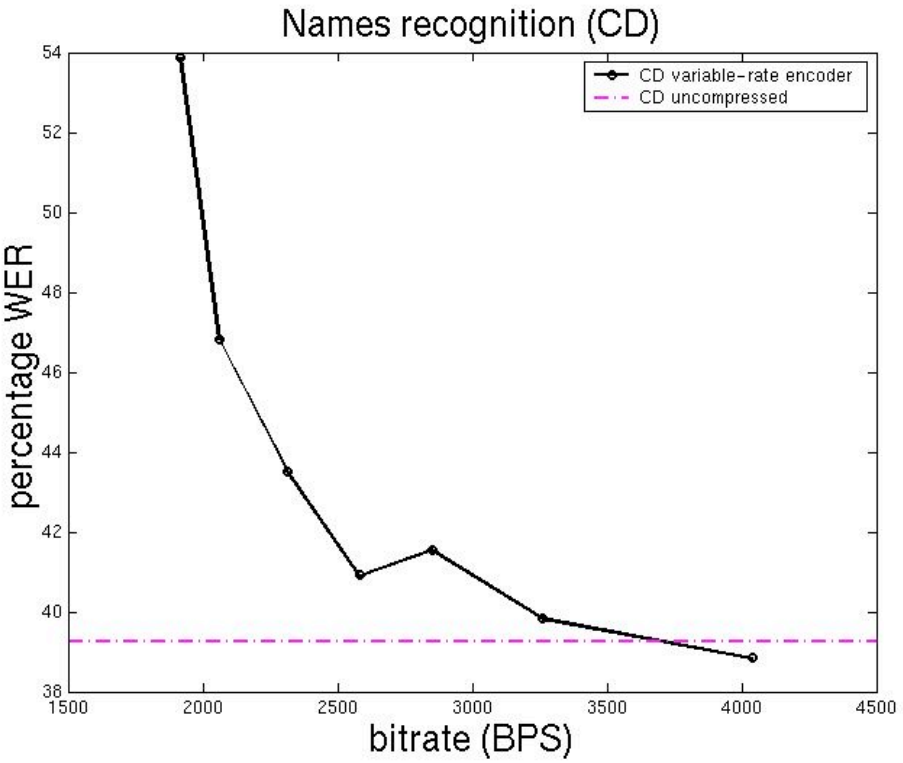


Figure 9: Names recognition results when CD models are used in the lattice recognizer.

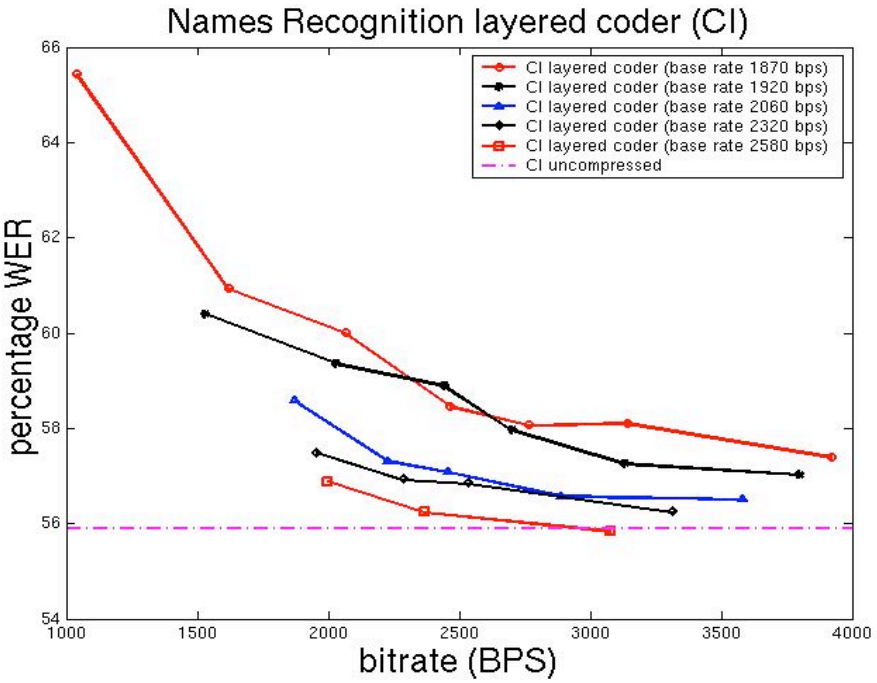


Figure 10: Names recognition results when the scalable encoder is used at the client and CI models are used in the lattice recognizer.

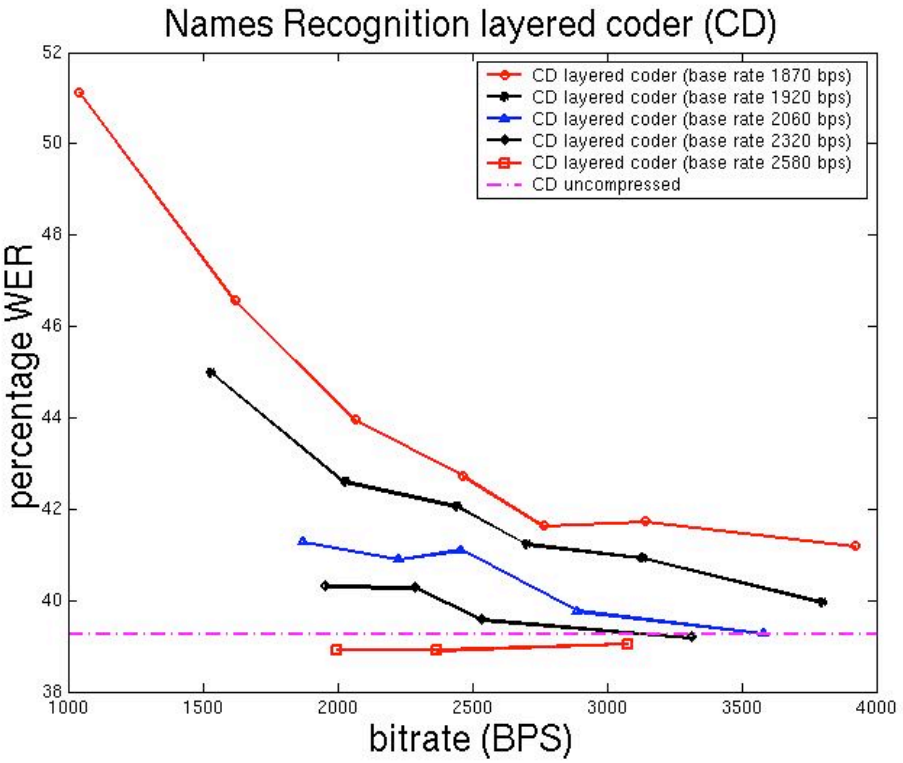


Figure 11: Names recognition results when the scalable encoder is used at the client and CD models are used in the lattice recognizer.

6. Other Relevant Work Being Conducted and How this Project is Different

There are a few papers published on the distributed recognition problem. Digalakis et al. [5] have shown that compressing the feature vectors that are used in speech recognition is effective. They evaluate uniform and non-uniform scalar quantizers, vector quantization and product-code quantization of the acoustic features and achieve bit rates between 2.6 kbps and 10.4 kbps.

Ramaswamy and Gopalakrishnan [6] also compress acoustic features used in speech recognition. In their method, correlations in frames of the acoustic features are exploited by linear prediction, and they use 2 stage Vector Quantizers to quantize the prediction errors. With this scheme they achieve a fixed rate of 4 kbps.

In our work [10], we use first order linear prediction and scalar quantizers to compress the acoustic features. The use of scalar quantizers as opposed to vector quantizers reduces the computational cost. More efficient design of scalar quantizers considering the importance of every element in the acoustic frame enables better compression performance. To quantize the acoustic features we use two different techniques, namely, entropy constrained scalar quantization (ECSQ) [7] and uniform scalar quantization (USQ). The proposed methods in [5.6] only exploit the redundancy and correlation in the acoustic feature data, and do not consider the classification properties of the speech recognizer. In our recent work [11] we have proposed efficient joint design of quantizers, which can work in conjunction with a complex classifier. The goal there was to minimize the classification error introduced by quantizing the data using

encoders operating on low dimensional inputs, which are subsets of the high dimension data used by the classifier for classification. This situation is analogous to the problem we are trying to solve here, i.e., we are independently quantizing every frame of the acoustic features, while all the acoustic feature frames are used for recognition. By using the techniques developed in [11] it is hoped that we can achieve even better rate-recognition tradeoffs.

7. Plan for the Next Year

As mentioned above, in order to improve the performance of our algorithms, we need to tightly couple the compression algorithm to the processes to be performed by the HMM-based recognizer. This will allow us to determine, based on knowledge of the HMM processing, what information in the acoustic frame is most important for classification and recognition. Traditional joint compression and classification schemes [8,9] have assumed that the dimension of the classifier and compressor are the same. However in the case of interest we are using scalar quantization of the features, while using an HMM which essentially performs a vector classification. This problem, where recognizer and compressor have different dimensions has hardly been studied, and we are already devoting a significant effort to solving it. Fundamentally, compression and recognition are similar processes, but they use very different cost functions (fidelity in reproduction versus accuracy in the recognition) and they operate at significantly different time scales (much larger scale for recognition, for example). Our ultimate research goal is to bridge the gap between these two problems, and provide ways of trading off compression and recognition performance.

In the short term we are investigating different metrics for the speech encoder design which are directly relevant to the HMM operation. In the long term we want to integrate the scalable speech compression techniques with the techniques developed when the dimension of the classifier and compressor are different to develop an integrated distributed recognition system.

We are currently testing music similarity metrics to determine how they relate to perceptually important features. Several undergraduate students are involved in this project and are already contributing. Progress in this area will eventually lead us to developing novel techniques for compression of the relevant features.

8. Expected Milestones and Deliverables

Extending on the work done before in this field, we have shown that the optimal solution to compressing speech for speech recognition has not yet been reached. The gain got by using the redundancy in the acoustic frames is limited. To get further performance improvements we will need to consider the properties of the speech recognizer while compressing the acoustic frames. We have also addressed the complexity and encoder variability issues in distributed speech recognition systems.

9. Member Company Benefits

No companies have directly supported this project, but we have been active in publicizing the work at conferences and to potentially interested companies. As an example, the graduate student

most directly involved in this project was a summer intern at Speechworks, where he worked on a related project. We have also discussed some of these ideas with researchers at Nuance and Panasonic.

10. References

[1] Lawrence R. Rabiner, "A tutorial on Hidden Markov Models and Selected Application in Speech Recognition", Proc IEEE, Vol. 77 No. 2, Feb 1989, pp 257-286.

[2] S.B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Acoust., Speech, Signal Processing, vol. ASSP-28, No. 4, 1980 pp. 357-366.

[3] Samuel Bayer, "Embedding Speech In Web Interfaces", Proc ICSLP, Philadelphia, PA, Oct, 1996, pp 1684-1688.

[4] M. Dietz, H. Popp, K. Brandenburg, R. Friedrich, "Audio compression for network transmission". Journal of the Audio Engineering Society, vol.44, (no1-2), Audio Eng. Soc, Jan.-Feb. 1996. pp 58-72.

[5] Vassilios V. Digalakis, Leonardo G. Neumeyer, and M. Perakakis, "Quantization of Cepstral parameters for Speech Recognition over the World Wide Web", IEEE Journal on Selected Areas in Communication, Vol. 17. No. 1 Jan. 1999, pp 82-90.

[6] Ganesh N. Ramaswamy, and Ponani S. Gopalakrishnan, "Compression of Acoustic Features for Speech Recognition in Network Environments," Proc. IEEE ICASSP, 1998 pp 977-980.

[7] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy Constrained Vector Quantization", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-37, pp. 31-42, January 1989.

[8] Keren O. Perlmutter, Sharon M. Perlmutter, Robert M. Gray, Richard A. Olshen, and Karen L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification", IEEE Transactions on Image Processing, vol.5, no.2, p. 347-60, February 1996.

[9] J. Li, R.M. Gray, and R. A. Olshen, "Joint image compression and classification with vector quantization and a two dimensional hidden Markov model," Proceedings of the 1999 IEEE Data Compression Conference, J. A. Storer and M. Cohn, Eds., pp. 23-32, Computer Science Press, March 1999.

[10] 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. IEEE International Conference on Multimedia and Expo 2000.

[11] Naveen Srinivasamurthy and Antonio Ortega, "Joint Compression-Classification with Quantizer/Classifier Dimension Mismatch," in Visual Communications and Image Processing 2001, San Jose, CA, January 2001.

[12] Hua Xie and Antonio Ortega, "Feature representation and compression for content-based image retrieval,'' in Visual Communications and Image Processing 2001, San Jose, CA, January 2001.

[13] Naveen Srinivasamurthy, Antonio Ortega and Shrikanth Narayanan, "Efficient Scalable Speech Compression for Scalable Speech Recognition", Eurospeech 2001, Aalborg, Denmark, September 2001.

[14] Raghavendra Singh and Antonio Ortega, "Erasure recovery in predictive coding environments using multiple description coding", 1999, IEEE Workshop on Multimedia Signal Processing
[15] Naveen Srinivasamurthy, Shrikanth Narayanan and Antonio Ortega, "Use of Model Transformations for Distributed Speech Recognition", ISCA ITR-Workshop 2001 (Adaptation Methods for Speech Recognition) Sophia-Antipolis, France August 2001
[16] Beth Logan and Ariel Salomon, "A Music Similarity Function Based on Signal Analysis", IEEE International Conference on Multimedia and Expo (ICME), August 2001.
[17] Jonathan T. Foote, "Content-Based Retrieval of Music and Audio," in C.-C. J. Kuo et al., editor, Multimedia Storage and Archiving Systems II, Proc. of SPIE, Vol. 3229, pp. 138-147, 1997.
[18] J.-C. Junqua, "SmarTspelL™: A multipass recognition system for name retrieval over the telephone," in IEEE Transactions on speech and audio processing, vol.5, pp.173--182, March 1997.
[19] P. Coletti and M. Federico, "A two-stage speech recognition method for information retrieval applications," in Eurospeech '99, (Budapest), September 1999.
[20] Y. Abe, H. Itsui, Y. Maruta, and K. Nkajima, "A two-stage speech recognition method with an error correction model," in Eurospeech '99, (Budapest), September 1999.
[21] A. Sethy, S. Narayanan, and S. Parthasarathy, "Syllable-based recognition of spoken names," in ISCA Pronunciation Modeling and Lexicon Adaptation Workshop, 2002.
[22] F. Behet, R. de Mori, and G. Subsol, "Very large vocabulary proper name recognition for directory assistance," in IEEE International Conference on Automatic Speech Recognition and Understanding 2001, pp.222 --225, 2001.
[23] Y. Gao, B. Ramabhadran, J. Chen, H. Erdogan, and M. Picheny, "Innovative approaches for large vocabulary name recognition," in ICASSP 2001, vol.1, pp.53--56, 2001.
[24] 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.
[25] Naveen Srinivasamurthy, Antonio Ortega and Shrikanth Narayanan, "Efficient Scalable Encoding for Distributed Speech Recognition," Submitted to IEEE Transactions on Speech and Audio Processing, January 2003.