Graceful Degradation of Speech Recognition Performance Over Lossy Packet Networks

Eve A. Riskin, Constantinos Boulis, Scott Otterson, and Mari Ostendorf

Department of Electrical Engineering
University of Washington, Seattle, WA 98195, USA
{riskin,boulis,scotto,mo}@rcs.ee.washington.edu

Abstract

This paper explores packet loss recovery in client-server Automatic Speech Recognition (ASR) systems. A forward error correction (FEC) system is designed and tested over several channel loss models, at variable amounts of data acquisition delay. In experiments with simulated packet loss, the FEC system provides robust ASR performance which degrades gracefully as packet loss rates increase. Comparing this scheme to several alternatives under low and medium loss channel conditions, we found one approach (multiple transmission plus interpolation) that yielded similar performance, but the FEC system should scale better to lower bit rate conditions.

1. Introduction

Because ASR systems require more computational resources than are currently available in hand-held wireless devices, a client-server ASR architecture is often proposed. In such a scheme, a simple, low power front end quantizes and packetizes speech, and sends it over a communication channel to an ASR server to perform speech recognition. Little attention in the ASR community has been paid to packet loss in the communication channel. In contrast, the image coding community has been studying the problem and numerous successful solutions that assign FEC to the coded image have been developed [8, 2, 3].

One approach to dealing with packet loss in ASR is to use interpolation to fill in the missing frames [7], but this may not be effective for bursts of loss. Here, we propose an alternative approach that accepts a small increase in transmission rate to introduce an error correction code. In this scenario, illustrated in Figure 1, a standard acoustic processor in the client generates MFCC vectors which are then quantized with a product vector quantization (VQ) code. The data are then organized into packets containing a mix of speech and FEC symbols. The packet groups, or "ensembles," are then sent through a communication channel to the ASR server. Packet loss recovery is accomplished by the decoder and then the data are restored to continuous-valued MFCC vectors by VQ codeword lookup. Speech is then recognized by a standard ASR algorithm.

In the sections that follow, we begin by providing background on the forward error correction approach used here, borrowed from image compression, and then extend the method to the ASR application. Then we describe experiments comparing error protection to various alternatives, showing significant performance gains in bursty loss conditions.

2. Approach

2.1. Forward Error Correction

FEC, such as Reed-Solomon coding [10], protects against erasures in data sent over lossy packet networks such as wireless networks or the Internet. In an \( (N, k) \) Reed-Solomon code, \( N - k \) symbols of FEC are used to protect \( k \) symbols of data, meaning that \( N \) symbols of data in total are transmitted. As long as any \( k \) of the \( N \) symbols are received, the \( k \) original symbols of data can be recovered.

In [8], an unequal loss protection (ULP) algorithm was designed to assign unequal amounts of FEC to compressed image data. The ULP algorithm was designed for embedded codes, which allow intermediate reconstructions of the image from any beginning part of the bitstream and hence it always assigns more FEC to the early parts of the coded bitstream. As a result, there is graceful degradation of compressed image quality with increasing packet loss; if a large amount of packet loss occurs, the decoded image is simply reconstructed from a shorter prefix of the coded bitstream, leading to reduced image quality instead of catastrophic failure.

When speech MFCC vectors are encoded using tree-structured VQ [1], the code is also embedded; thus, the ULP algorithm can be conveniently applied to speech data. The open question is how much FEC should be assigned to the MFCC vectors. Whereas the work in [8] optimized a peak signal-to-noise ratio, here we will assign FEC to minimize word error rate (WER). Similar to the image coding work, when packet loss occurs, the WER performance will degrade gracefully instead of falling off sharply. Thus, useful ASR will be possible.


In all of our experiments, the subvectors were individually quantized using binary tree-structured VQ with fixed length coding. Euclidean distance was used as the distortion criterion [5]. As it was shown in [4] and confirmed in our case, we can quantize the cepstrum parameters at about 2.5 Kbps and have no degradation in ASR performance relative to the original cepstral vectors.

2.3. Bit Allocation by Pruning

After the subvectors are determined, we next construct a graph of WER vs. number of bits lost, referred to as the pruning curve. Starting from an initial (source) bit rate we delete bits one by one so as to have the smallest possible increase in WER after each deletion. This is an optimal inverse to the bit allocation assignment done in [4].

The optimum trade-off of WER and bit rate can be obtained by evaluating all possible pruning combinations and picking the one with the lowest WER. However, this exhaustive search demands huge amounts of processing time. Therefore we experimented with various pruning criteria which are orders of magnitude less expensive and would hopefully correlate well with WER. We tried the Euclidean distance, the Mahalanobis distance using the global covariance matrix, and the acoustic likelihood of the training data. At each stage of the pruning process, we selected the $M$ best candidates and ran $M$ WER experiments, keeping the best one.

Another approach, which leads to better results, is to use an $N$-step lookahead pruning algorithm using WER as the pruning criterion. This procedure breaks the problem into sets of size $N$ each, performing exhaustive search in each set. We generate pruning curves by starting at high bit rates, where the initial allocation of bits to subvectors is not important because the performance has already converged to the continuous case and minor changes do not impact WER. We experimented with $N = 1, 2, 3, 4$, but the pruning curves were found to be almost identical. We also experimented with different starting points (4Kbps and 2.8Kbps), and the results were insensitive to these changes, too. After running the bit allocation algorithm, we obtained a trajectory of number of bits vs. WER. This trajectory is stored and used later during the FEC symbol assignment.

2.4. FEC Assignment

In this work, we packetized our speech recognition data into packet sizes and data rates found in Internet VoIP. Specifically, we chose 10 byte packets, as found in G.729, and limited the data rate of the speech data plus FEC symbols to a total of 4800 bps, as in some LPC vocoders. Having fixed these parameters, the remaining choices are the size (in packets) of each speech+FEC “ensemble,” and the number and location of bits in each speech/FEC symbol. These choices were made by imposing the constraint that the ensemble will contain no more than 20 MFCC vectors, limiting data acquisition delays to 200ms. Having defined an ensemble, the next task is to determine the FEC assignment which minimizes the expected WER. Ideally, this would be done by iteratively testing candidate FEC assignments, running ASR with a Monte Carlo channel loss model.

---

### Table 1: Cepstrum coefficients assigned to each subvector.

<table>
<thead>
<tr>
<th>Subvector</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC Coeffs</td>
<td>(0,1)</td>
<td>(2,3)</td>
<td>(4,5)</td>
<td>(6,7)</td>
<td>(8)</td>
</tr>
</tbody>
</table>

---

Figure 2: Subvector bit assignment.

Figure 3: FEC assignment after running optimization algorithm.
Since this is computationally prohibitive, we instead estimate the expected WER for a given FEC allocation by using the WERs stored when pruning was performed. Under this method, the FEC assignment is chosen by minimizing

$$\left( \sum_{k=1}^{K} p(k) \right)^{-1} \left( \sum_{k=1}^{K} p(k) \mathcal{E}(B(k)) \right)$$

over all possible FEC assignments, where $K$ is the number of packets in the ensemble, $p(k)$ is the probability of losing $k$ packets according to the exponential channel model, $B(k)$ are the bits which are available when $k$ packets have been lost and $\mathcal{E}(B(k))$ is the word error rate seen during the bit allocation process when MFCC vectors were quantized with $B(k)$.

There are two effects not accounted for here. First, the exponential packet loss model does not model the temporal loss dependency (burstiness) of real communication channels. Second, during the bit allocation process, the WER experiment was performed with all vectors being quantized at $B(k)$; in a real channel, each ensemble could have a different quantization, depending upon the packet loss. While this method does not model all packet loss effects, we have found that it does provide a reasonable cost function for FEC assignment optimization.

Since our packets and ensembles were small, it was possible to simply search over all FEC assignments for that which yielded the best estimated expected WER.

3. Experiments

3.1. Test Conditions

All the experiments were carried out on a corpus of speech from human-computer dialogs about air travel information, part of the DARPA Communicator program. The data we had available were collected over the standard telephone. To evaluate our algorithms we used the Nuance 7.0 recognizer with the generic telephony acoustic models distributed by Nuance. We trained trigram language models using about 50,000 sentences of Communicator data. For evaluation purposes we used a test set consisting of 2096 words, collected at Carnegie Mellon University. The WER for this test set is 22.9%, comparable to what other labs are reporting [9].

Although the FEC assignment was optimized with an exponential packet loss probability distribution function, most real channels exhibit burstiness. We have modeled such channels by using a 2-state Markov model [6], known also as a Gilbert model. In Figure 4, $p$ is the probability that the next packet is lost, provided the previous one has arrived; $q$ is the opposite. If $p+q = 1$ the Gilbert model reduces to a Bernoulli model. $q$ can be seen as controlling the burstiness of packet losses. The tests were run under the loss conditions reported in [12, 6] and are summarized in Table 2, where $udp = p/(p + q)$ is the unconditional loss probability which was used in the FEC assignment exponential model, and $cep = 1 - q$ is the loss probability, conditioned on the event that the previous packet was lost.

### Table 2: Channel Loss Test Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$clp$</td>
<td>0.147</td>
<td>0.33</td>
<td>0.50</td>
<td>0.60</td>
</tr>
<tr>
<td>$udp$</td>
<td>0.006</td>
<td>0.09</td>
<td>0.286</td>
<td>0.385</td>
</tr>
</tbody>
</table>

3.2. Alternatives to FEC

We compared our FEC approach to a number of alternatives. For the worst case baseline, we replace the missing frames with the mean frame over all training data. The position of the missing packets can be identified by the sequence id number which is included in the header of each packet. With 10-byte packets and 26 bits per frame, the loss of one packet implies that 3 frames are lost.

The alternative is interpolation: when a frame is lost, it is linearly interpolated using the neighboring frames. The weight of each one of the neighbors is also a linear function of the distance of the missing frame from its most recent correctly received neighbors. This interpolation scheme was proposed in [7] and for the task to which it was applied, it worked surprisingly well. However, in [7], each frame is assumed to fit in one packet (a very inefficient packetization scheme) and the channel is not bursty. In our case, losing a packet results in the loss of 3 consecutive frames and also the loss pattern of the packets can be extremely bursty.

The third alternative is a multiple transmission scheme. In a sequence of 6 frames, the baseline system will use 2 packets, where the first packet will have frames $\{1, 2, 3\}$ and second packet will have frames $\{4, 5, 6\}$. In the multiple transmission method a sequence of 6 frames is packetized using 4 packets. Packet 1 will have frames $\{1, 2, 3\}$, packet 2 will have frames $\{2, 3, 4\}$, packet 3 will have frames $\{3, 4, 5\}$ and packet 4 will have frames $\{4, 5, 6\}$. This means that we essentially double the transmission bit rate. It is similar to transmitting each of the baseline packets twice, with the difference that the center frames (frame 3 and 4) are repeated more times than the start/endpoint frames and so they will be received with higher probability. When combined with interpolation this scheme is expected to perform better than either multiple transmission or interpolation alone.

3.3. Test Results

Figure 5 shows the measured WER for each of the channel conditions in Table 2. Another point illustrated by Figure 5 is that the WER generally decreases with increasing delay, although the decrease is not monotonic because not all ensembles are robust. For example, all of the constraint-meeting ensembles at a delay of 80ms had bit rates of 4000 bps, while there is a 4800 bps ensemble with a delay of only 50ms (the low delay means that there are few data bits in the ensemble; the rest is FEC).

Table 3 summarizes the best WER results for each method. Under these conditions, both the FEC and the multiple transmission with interpolation schemes are robust to packet loss. Other methods show severe performance degradation, particularly for the most bursty condition. The FEC approach requires a slightly lower bit rate (4.8Kbps vs. 5.2Kbps), but incurs more delay (200ms for all cases vs. 40ms for each frame lost). We
have also included Forward Error Correction with equal loss protection (ELP). The results clearly show that assigning more FEC to more important subvectors leads to substantial gains.

### 4. Conclusions and Future Work

In this paper an FEC scheme for reducing the effects of packet loss was specifically tailored for ASR applications. In experiments using simulated packet loss, it was shown that with 4.8 Kbps and tolerable delay, the degradation in performance can be kept small even under extremely adverse network conditions. The FEC scheme compares favorably to interpolation and multiple transmission, but the combination of these two techniques is competitive (requires higher bit rates, but incurs less delay). This result shows that even when quantizing the speech frames with 2.6 Kbps there is still much redundancy.

We conjecture that with a better compression method we will not only lower the bandwidth required but that this will result in a bigger differential in WER between FEC and the alternatives. One possibility under investigation is joint quantization of successive frames. Preliminary experiments have shown that the mutual information between the same coefficients of neighboring frames is 10 times higher than that of coefficients of the same frame. Such a strategy would require new packet allocation schemes to avoid extra delay.

In future work, we plan to alter the training of the quantizer. Here, minimum Euclidean distance was used as the training criterion for the quantizer. A more suitable criterion (but still practical) would be maximum likelihood. Also, in this work we have used continuous (Gaussian mixture) distribution models. Given that the cepstral features are quantized, it can be more cost-effective to use discrete mixture models as in [4].

### 5. Acknowledgements

We gratefully acknowledge the help and advice of Sarah Schwarm, Alexander Mohr and Petros Gavalakis. We also thank the speech group at CMU for providing Communicator data that was used in this study. This work was supported by DARPA, contract number N660019928924 and NSF Grant No. EIA-9973531.

### 6. References


