MITIGATION OF CHANNEL ERRORS IN EFR-BASED SPEECH RECOGNITION

Ángel M. Gómez, Antonio M. Peinado, Victoria Sánchez, José L. Pérez-Córdoba, Antonio J. Rubio

Dept. of Electronics and Computer Technology
Universidad de Granada, Spain
amgg@ugr.es

ABSTRACT

Network-based speech recognition (NSR) using the conventional speech channel with the Enhanced Full Rate (EFR) or the Adaptive Multi-Rate (AMR) codec is a very attractive approach since no change to existing mobile phones is needed. However, NSR reveals a degrading performance due to both transmission channel errors and the speech encoding process in comparison with Distributed Speech Recognition (DSR), where speech features are efficiently coded and transmitted on a data channel.

In this paper we focus on the degradation of the speech features caused by channel errors in an NSR system and propose methods to improve the quality of these features. Applying these methods, it turns out that the performance of an NSR system based on EFR coding is comparable to that based on DSR.

1. INTRODUCTION

The increasing development of cellular networks has thrown down a new challenge: the speech recognition in mobile devices that enables the access to voice activated services. These services can be implemented in a variety of conceptual solutions. A first approach could be to perform the speech recognition in the mobile device itself. Although this embedded solution can be feasible, its functionality is quite limited by hardware constraints and power consumption. It is therefore considered to be more efficient and practical to perform the recognition on a remote server.

In this scenario, there are two approaches. The first one, widely employed today, is known as *Network-based Speech Recognition* (NSR) [1]. NSR uses a full-duplex speech channel, with speech coding (for bit rate reduction) and channel coding (for error protection), to send the speech data to the remote server. The second one is referred to as *Distributed Speech Recognition* (DSR) [2]. In DSR, the speech recognition task is distributed between the local mobile device, which extracts and encodes the speech features, and the remote server, which performs the recognition itself. In this way, DSR avoids the speech coding step and the transmission is performed over a data channel, unlike in NSR.

However, there are still some problems in the final deployment of DSR. The current handsets are not capable of carrying out feature extraction and it would be necessary to include new hardware in the device. In addition, for some types of applications, it would be desirable to have the transmitted speech signal available just in case a further verification is required. Finally, although a standard has been established to ensure compatibility between the terminal and the remote recognizer [2], it does not cover the areas of data transmission or any higher level application protocols needed for the final implementation.

The NSR approach avoids these problems by performing the recognition from the decoded speech. The Enhanced Full Rate (EFR) codec, the most widely used codec in GSM, can achieve very similar results to DSR with clean transmission [1,3]. However DSR outperforms it in the presence of channel errors. In this work, we analyze these errors and their effects on speech recognition (section 2) and propose some solutions for them (section 3). Finally, the conclusions of this work are summarized in section 4.

2. EXPERIMENTAL FRAMEWORK

In order to evaluate and compare the techniques proposed in this paper, the ETSI STQ-AURORA Project experimental framework was adopted [4]. The speech data has been extracted from clean sentences of the Aurora-2 database (connected digits spoken by American English speakers). Training is performed from a set of 8440 clean utterances and test is carried out over the clean sentences of set A, with 4004 utterances.

The front-end used in this work is the one proposed in the ETSI standard [2]. It provides a 14-dimension feature vector containing 13 MFCCs (including the 0th order one) plus log-energy. The recognizer is the one provided by Aurora and uses eleven 16-state continuous HMM word models, (plus silence and pause, that have 3 and 1 states, respectively) with 3 Gaussians per state (except silence, with 6 Gaussians per state). The recognition performance is measured in terms of word accuracy.

Under the EFR scheme, the speech samples are transmitted using a full-duplex channel. These samples are coded and decoded according to GSM 6.60 standard [5]. Channel coding, decoding, error detection and correction and bad frame mitigation tasks are accomplished according to GSM 5.03 and GSM 6.61 [6,7]. On the other hand, under the DSR scheme, the speech features obtained from the front-end are quantized using a Split Vector Quantizer (SVQ) that groups them into pairs (MFCCs 1 and 2, MFCCs 3 and 4, ..., MFCC 0 and log-Energy). Each pair has its own codebook that is generated utilizing a weighted distance measure. The resulting bitstream is transmitted according to ETSI DSR standard through a data channel. After decoding, the error mitigation algorithm proposed in the DSR ETSI standard is applied.

Work supported by the Spanish CICYT Project TIC-2001-3323

The channel is simulated using the GSM error patterns (EPx, *x=1,2,3*) to corrupt the bit stream. These error patters are in AEG format and represent three channel conditions: EP1 (10dB C/I, good quality), EP2 (7dB C/I, medium quality) and EP3 (4dB C/I, lower quality).

3. ANALYSIS OF GSM-EFR CHANNEL ERRORS

When a speech frame reaches the receiver, it is decoded applying error correction and checking the various protection mechanisms included in the frame. As a result of this process, a Bad Frame Indicator (BFI) will be enabled if a transmission error is detected in that frame. Due to the discriminative treatment of the frames by the decoder, we should distinguish between different types of noises derived from the channel noise.

3.1. Bad frame noise and background noise

If the BFI of a frame is enabled (BFI=1), that frame has been seriously damaged and its synthesis would be very unpleasant for a listener. For this reason, in order to improve the subjective perception of the signal, these frames are replaced by a repetition or extrapolation of the last received frame or frames. The GSM standard 6.61 does not impose any specific substitution and muting algorithm, however, it proposes an example which is usually the implemented one [7]. This substitution is performed in such a way that the output level gradually becomes comfort noise. On the other hand, it can not be asserted that those frames whose BFI are not marked (BFI=0) are entirely correct, since the error protection is perceptually applied and all the speech parameters are not equally protected. In this sense, the signal transmitted with errors can present anomalies with regards to the clean transmitted signal, which would influence the recognition accuracy.

In our study, we have designed two experiments intended for evaluating the effects of the aforementioned errors. We have tested the system performance in these situations:

- *Background Noise*. In this situation, we only take into account the noise generated by unmarked frames (BFI=0). The frames marked as no valid (BFI=1) are replaced with the corresponding valid frame. This valid frame is built up from the parameters obtained in a clean transmission.

- *Bad Frame Noise*. In this case, we only take into account the noise from frames with marked BFI. In order to do this, all frames, except those whose BFI is enabled, are replaced with those obtained in a clean transmission. This avoids the presence of anomalies in unmarked BFI frames.

Table 1 (cols 4 and 5) shows the results obtained from these two experiments together with those of DSR and EFR (cols 2 and 3), under different channel conditions. As it can be seen, the *Bad frame noise* is mainly the responsible for the performance degradation in noisy channel conditions, while the *Background noise* has a negligible importance. Due to the bursty nature of the wireless channel (in spite of the interleaving introduced in the encoder) a high proportion of bad frames appear consecutively, that is, constituting bursts. The recognition accuracy in accordance with the length (*l*) of the bursts on EP3 condition is shown in figure 1. When *l<1*, there are no bad frames (a burst involves at least 1 bad frame), and we obtain the same results as in clean conditions. In the other extreme, when the length is less than ∞, all bad frames are present and we obtain the same results as with the EP3 condition. As it can be observed, the incremental accuracy reduction is negligible for burst lengths bigger than five, due to the small frequency of appearance of these bursts.

Channel	DSR	EFR	Backgr. Noise	Bad Fr. Noise	Codec Memory Noise	Bad Fr. Isolated Noise
Clean	99.04	98.70	-	-	-	-
EP1	99.04	98.44	98.50	98.61	98.44	98.68
EP2	98.95	96.91	98.31	97.53	97.73	98.28
EP3	93.41	84.48	98.22	85.80	93.54	90.47

Table 1. *Word accuracy in recognition with each kind of noise.*

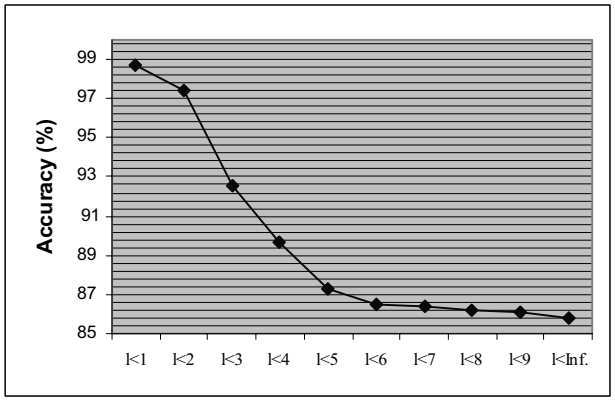

Figure 1. *Word accuracy versus burst length (l) on EP3.*

3.2. Codec memory noise

Analyzing the speech samples obtained in a noisy transmission, we can observe a degradation of the signal corresponding to correct frames after a burst (bad frame noise). This is due to the memory of the CELP type codec. In this sense, although several frames after the burst had been received without errors, the resulting signal would be degraded due to the previous erroneous frames. This degradation constitutes what we call *Memory noise*.

In the same way as in the previous experiments, we can isolate the effects of the bad frame noise from the memory noise. In this case, we previously eliminate the background noise and operate at a different level substituting speech samples instead of speech parameters. We consider two new experiments:

- *Bad Frame Isolated Noise*. In this case, we isolated the alterations caused by the bursts, not by the associated memory effect. To this end, speech samples belonging to good frames (BFI=0) are replaced with the corresponding correct speech samples.

- *Memory Noise*. In this experiment, the noise generated after a burst by the memory of the codec is isolated. For this purpose, speech samples belonging to bad frames (BFI=1) are replaced with the corresponding correct speech samples, leaving only the corruption corresponding to memory noise.

Table 1 (cols 6 and 7) shows the results of both experiments. As it can be seen, the memory noise has an important responsibility for the reduction of the recognition accuracy. This confirms the influence of a burst beyond its limit in terms of bad frames.

4. IMPROVING ACCURACY OVER EFR

Since our goal is the recognition of the speech degraded by channel errors, we will try to compensate the speech features used for recognition rather than the speech signal. However, there are differences of size and shift between the windows of the EFR codec and the Aurora feature extractor. This difficulty can be avoided with a mapping function which relates each bad EFR frame with a bad feature vector. In our work, this function is defined as:

$$F_{map}(n)=\begin{cases}1 & \left(BFI\left(\left\lfloor\frac{n}{2}\right\rfloor\right)=1\right) or \left(BFI\left(\left\lfloor\frac{n+1}{2}\right\rfloor\right)=1\right) \\ 0 & ;otherwise\end{cases} \quad (1)$$

where n is the time index of a given feature vector and $BFI(m)$ is the bad frame indicator of frame m (both starting from 0). $F_{map}(n)$ is 0 when feature vector n is received and equal to 1 when feature vector n is bad (see figure 2).

4.1. Burst reconstruction

Whenever an error burst appears, frames with BFI enabled are considered as lost frames in GSM 6.61 [5]. In this situation, there is no information about the original signal. Then, the corresponding bad feature vectors (according to the mapping function) are lost and must be reconstructed. This reconstruction can be accomplished from the last and the first vectors received before and after the burst, respectively, by means of a simple linear interpolation:

$$\hat{\boldsymbol{x}}(t)=\boldsymbol{x}(t_s)+\frac{\boldsymbol{x}(t_e)-\boldsymbol{x}(t_s)}{t_e-t_s}\left(t-t_s\right) \quad \left(t_s<t<t_e\right) \quad (2)$$

where $\hat{\boldsymbol{x}}(t)$ is the estimated feature vector at time t, $\boldsymbol{x}(t_e)$ is the first vector after the burst and $\boldsymbol{x}(t_s)$ is the last vector before the burst. Although this is a very simple technique, an important improvement over EFR can be obtained as shown in table 2 (col 4, *EFR interpolation*).

4.2. Memory noise compensation

By contrast to burst errors, where there are lost frames, the memory noise only involves signal degradation. In a first approach, this noise can be considered similar to acoustic noise, whereby, under this assumption, it is feasible to apply an acoustic noise compensation algorithm as FCDCN (Fixed Codeword-Dependent Cepstral Normalization) [8] to try to compensate the codec memory noise. FCDCN applies a correction based on simultaneously recorded noisy and clean speech data (stereo data). This correction depends on the instantaneous SNR (Signal-to-Noise Ratio) of the input. Furthermore, for each codeword q, we should consider a different correction. It usually represents a quantization index for the input, relating the speech vectors of the input and the correction factor to apply. In this way, the degraded speech vectors are compensated as follows:

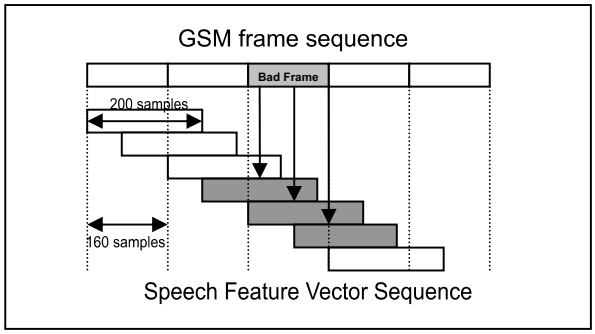

Figure 2. *Mapping function between GSM frames and speech feature vectors.*

$$\hat{\boldsymbol{x}}=\boldsymbol{x'}+\boldsymbol{r}\left[SNR,q\right] \quad (3)$$

where $\hat{\boldsymbol{x}}$ is the estimated vector, $\boldsymbol{x'}$ is the noisy vector and $\boldsymbol{r}$ is the correction vector for a given SNR and a given codeword q.

The memory noise depends on the previous burst error: the longer the burst length, the higher the noise level. Furthermore, it decreases as good frames are received. Therefore, the instantaneous *SNR* of the noisy feature vector depends on the previous burst length and the distance to it. Due to this dependence, a different FCDCN correction should be applied for each burst length and time interval after it, modifying equation (3) to:

$$\hat{\boldsymbol{x}}=\boldsymbol{x'}+\boldsymbol{r}\left[l,d,q\right] \quad (4)$$

where l is the length of the previous burst and d is the distance from the current noisy vector $\boldsymbol{x'}$ to the first vector after the burst (measured in number of vectors).

In order to reduce the computational burden and memory requirements, the maximum burst length can be limited to $l=5$, since the reduction on accuracy is negligible for longer burst lengths (figure 1). Furthermore, the maximum distance to the burst can be set on $d=20$ feature vectors after the burst. As far as we know, this is a safe limit since the correction factors at that distance are close to zero (no correction).

On the other hand, the codewords are defined through the SVQ quantizer used by the Aurora front-end described on section 2. This involves working with feature pairs instead of feature vectors. In this way, each noisy feature vector is quantized giving seven indices or codewords. Each one selects the compensation factor for its corresponding feature pair. This gives the last modification on equation (3):

$$\hat{\boldsymbol{p}}=\boldsymbol{p'}+\boldsymbol{r}(l,d,q) \quad \left(q=SVQ(\boldsymbol{p'})\right) \quad (5)$$

where $\hat{\boldsymbol{p}}$ is the compensated feature pair while $\boldsymbol{p'}$ is the noisy feature pair and q is the SVQ quantization index for $\boldsymbol{p'}$.

Every feature vector received after a burst (affected by memory noise) is compensated until its distance to the previous burst is longer to 20 or a new burst appears. On the other hand, if the previous burst length is longer than 5, we reuse the correction factors applied on burst lengths equal to 5.

Finally, this algorithm requires stereo speech data in order to compute the compensation factors. The training database can be used to accomplish this objective. Simulating burst errors of same length during the transmission, we can build up as many stereo data as it is needed. Given M bursts with the same length l, each one ending at time t_n ($n=1,..,M$), the correction factors at distance d are calculated as follows:

$$\begin{aligned} \boldsymbol{r}(l,d,q) &= \frac{1}{M}\sum_{n=1}^{M}\left(\boldsymbol{p}_{t_n+d} - \boldsymbol{p'}_{t_n+d}\right) & 1 \le l \le 5 \\ q &= SVQ\left(\boldsymbol{p'}_{t_n+d}\right) & 0 < d \le 20 \end{aligned} \qquad (6)$$

where $\boldsymbol{p}_t$ and $\boldsymbol{p'}_t$ are the clean and noisy feature pairs at time t ($t=t_n+d$), respectively.

The proposed adapted FCDCN for memory noise compensation can be used together with linear interpolation for burst reconstruction. Due to the fact that the burst reconstruction depends on the previous and next received vector, it must be applied after memory noise compensation. Table 2 shows the results obtained applying both algorithms (col 5, *EFR Interp. & Adapted FCDCN*). It can be seen that it outperforms EFR, approaching the DSR performance.

4.3. Extension to codec noise

The aforementioned correction factors were computed comparing noisy transmitted speech with clean transmitted encoded speech. However, we can also compensate the distortion introduced by the coding process by computing the correction pairs from non-encoded speech. In this way, memory and codec noise are compensated at the same time after every burst. Moreover, an additional set of correction pairs, $\boldsymbol{r}(q)$, can be computed comparing encoded and non-encoded speech. This set is applied over the feature vectors in the beginning, when there is no previous burst, and after the 20 vectors after a burst, compensating only the distortion introduced by the codec.

Table 2 shows the results of this extension combined with linear interpolation for burst reconstruction (col 6, *EFR Interp. & Adapted FCDCN (clean speech)*). Extending in this way the algorithm, we can improve the recognition accuracy, obtaining a performance quite close to DSR one.

5. CONCLUSIONS

In this work, we have focused our study in the effect that trasmission channel errors have on a NSR system using the EFR speech codec. We have analyzed the impact of three different types of errors over the recognition system caused by an erroneous transmission: background noise, bad-frame isolated noise and codec memory noise.

From this analysis, we have observed that the system degradation is mainly due to the two last error types. By means of a differentiated treatment of these errors, we have shown that the NSR approach can achieve results similar to DSR. Furthermore, the proposed algorithms do not have especial computational requirements and can be easily applied on real time. Besides, they are extensible to other CELP type codecs.

Channel	DSR	EFR	EFR Interpolation	EFR Interp. & Adapted FCDCN	EFR Interp. & Adapted FCDCN (clean speech)
Clean	99,04	98,70	98,70	98,70	98,81
EP1	99,04	98,44	98,43	98,46	98,64
EP2	98,95	96,91	97,55	97,82	98,19
EP3	93,41	84,48	90,76	94,04	94,04

Table 2. *Word accuracy (%) in recognition with the proposed enhancement algorithms.*

The main disadvantage of these algorithms is the necessity of the BFI flags. The mapping function requires these flags to discriminate between received and lost speech frames. This requires either direct access to the GSM bitstream or an algorithm capable of directly detecting bad frames from the speech samples.

Finally, in the case of the AMR codec, the speech and channel encoding is modified in order to face the channel conditions. Although less affected by transmission channel errors, we will still have similar errors as the ones described in this paper. In AMR we would additionally have to consider the degradation introduced by the speech encoding process on the NSR system. Further work will address this problem.

6. REFERENCES

[1] T. Fingscheidt, S. Aalburg, S. Stan and C. Beaugeant, "Network-based vs. distributed speech recognition in adaptive multi-rate wireless systems", ICSLP 2002, Denver, September 2002.

[2] "ETSI ES 201108 Speech Processing, Transmission and Quality aspects; Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", ETSI Standard, 2000.

[3] H.G. Hirsh, "The influence of speech coding on recognition performance in telecommunication networks", ICSLP 2002, Denver, September 2002.

[4] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions", *ISCA ITRW ASR2000*, Sept. 2000.

[5] "ETSI EN 300 726. Enhanced Full Rate (EFR) speech transcoding", ETSI Standard, 1999.

[6] "ETSI EN 300 909. Channel Coding", ETSI Standard, 1999.

[7] "ETSI EN 300 727. Substitution and muting of lost frames for Enhanced Full Rate speech traffic channels", ETSI Standard, 1999.

[8] A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition", Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon Univ., 1990.