SOFT DECODING STRATEGIES FOR DISTRIBUTED SPEECH RECOGNITION OVER IP NETWORKS

Antonio Cardenal-López, Laura Docío-Fernández and Carmen García-Mateo
Departamento de Teoría de la Señal y Comunicaciones
University of Vigo, Spain
{cardenal,ldocio,carmen}@gts.tsc.uvigo.es

ABSTRACT

In Distributed Speech Recognition, speech feature vectors are obtained at the client side, and transmitted to the remote server for recognition. In this paper, we investigate the robustness of the remote recognizer against the inherent packet loss in an Internet communication. In the decoding process, we propose to apply techniques used for "missing data" problems. The idea is to use a simple approach of error concealment to recover the non-received speech frames, and then to consider these recovered speech frames as not completely reliable. Thus, at recognition stage, our recognition engine uses a weighted (or soft decision) Viterbi algorithm in order to take into account the reliability of the recovered speech frames. Results on Aurora databases show that the proposed approach provides good recognition performance over a wide range of network conditions.

1. INTRODUCTION

The increasing use of both the Internet and Automatic Speech Recognition (ASR) systems makes Internet-based Distributed Speech recognition (DSR) services very attractive. These services are based on a client-server architecture: (1) the client device quantizes and packetizes the speech features and transmits them over the communication channel to a remote ASR server; (2) the remote ASR server performs the speech recognition task.

On congested IP networks, routers discard packets if their packet in-flow exceeds their out-flow for a given data route. This fact makes packet loss a key factor to take into account when developing a DSR system over IP networks, since complete segments of the speech signal can be lost.

In [1] and [2] we showed how packet loss affects the speech recognition performance and the effectiveness of simple error concealment techniques. We showed that the level of packet loss has a detrimental impact on the recognition performance and the repetition concealment technique is advantageous as far as isolated single losses are concerned, but as for bursty losses,this packet recovery approach is not useful.

We propose here two different decoding approaches that take into account whether the current speech frame has been received by the server (reliable), or its spectrum has been recovered by the recognizer (unreliable). These decoding approaches are based on a soft decision Viterbi algorithm, which includes a weighting factor on the probability of observation of the speech frame. This factor can be the same for any speech frame, or can be dependent on its position in the burst. This soft decision Viterbi decoder is proven to outperform the hard decoding in terms of recognition performance.

This work has been partially supported by the Spanish projects Transcrigal (TIC2000-1104-C02-01) and ITACA (TIC2002-022-08).

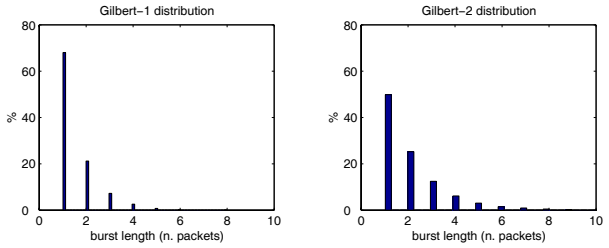

Fig. 1. *Burst length distribution for each Gilbert network condition.*

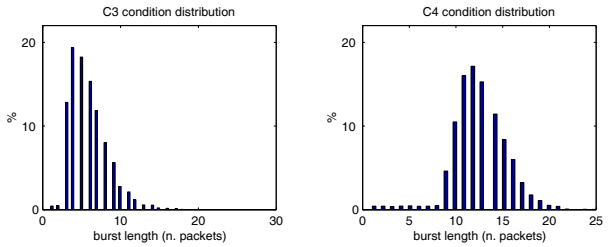

Fig. 2. *Burst length distribution for each Multi-state network condition.*

2. EXPERIMENTAL FRAMEWORK

2.1. Speech databases and Front-End

We have considered the ETSI STQ-AURORA Project Database 2.0 and 3.0 [3]. The client front-end is the advanced front-end proposed by the ETSI AURORA WI008 standard [4]. We consider a pair of speech frames as specified in [4] as basic unit (packet) for transmission over the IP channel.

2.2. IP network scenarios

Packet loss in an IP network occurs as a result of congestion in the network and the occurrence of packet loss is burst-like in nature, they are not independent on a frame-by-frame basis.

In order to measure the influence of missing speech packets on the ASR system performance, we have considered two different packet loss models as IP network scenarios.

2.2.1. Gilbert model

As a first approximation, we have simulated the IP network by using the known Gilbert model [5]. This model has two states: "Good or no loss" state and "Bad or packet loss" state. At any time, the probability of the next state is determined by only the current state and has no relationship with any previous state.

For this network scenario, the tests were run under the loss conditions presented in [1]. Figure 1 shows the distribution of the

Cond	NEP	NFR
C1	2.41	1.12
C2	13.40	2.55
C3	19.93	7.61
C4	30.88	21.62

Table 1. *Reference results for Aurora 2 (WER). NEP= No error protection, NFR = Nearest Frame Repetition. WER=1.01 with no losses.*

burst lengths. The examination of this figure reveals that for the Gilbert-1 distribution (C1 condition), the losses are predominantly solitary packet losses, and for the Gilbert-2 distribution (C2 condition) 90% of the bursts have a length in the range 1–3 packets.

2.2.2. Multi-state Markov model

In [1] we have also considered an IP network with a bottleneck topology. We have shown that in this scenario, the burst length distribution is very different from the one given by the Gilbert model. In order to approximate to this IP network scenario, we have considered an extension of the Gilbert model in which the "Bad or packet loss" state uses a k-state Markov chain to model the burst length, where the parameter k determines the minimum burst length.

Figure 2 shows the distribution of the burst lengths when the Bad state is modeled as a 3-state (C3 condition) and a 9-state (C4 condition) Markov chains. The examination of this figure reveals that in the 3-state case, 90% of the bursts have a length in the range 3–9 packets, the mean burst length being 6 packets. In the 9-state case, 90% of the bursts have a length in the range 9–16 packets, the mean burst length being 13 packets.

2.3. Back-End recognizer

We use as back-end recognizer, the one developed in our research group [6]. This decoder is based on two stages: (1) a Viterbi algorithm which works in a synchronous way with a beam search; and (2) an A^* algorithm. This recognizer was developed for large vocabulary continuous speech recognition applications.

The recognizer uses the word models and grammar as described in [7]. Before the decoding task, it also covers the error mitigation algorithm proposed in the Aurora standard [4]. In our earlier work [1][2], we showed that the performance of this packet recovery approach degrades rapidly with the increase in burst length. In the next section we present two different approaches that assign a reliability factor to the recovered speech frames, and thus, the recognizer uses a soft decoding Viterbi algorithm in order to improve the recognition performance.

3. SOFT-DECODING APPROACHES

3.1. Observation probability weighting

Presented in [8] is an approach that modifies the Viterbi recognizer to take into account the confidence in the decoded feature. The algorithm proposes to weigh the probability of observing the decoded feature given the HMM state model $b_j(\mathbf{o}_t)$ with the probability of decoding the feature vector $\mathbf{o}_t$ given the received value $\mathbf{y}_t$. Following this idea, we include a weighting coefficient γ_t ($0 \leq \gamma_t \leq 1$), which is a confidence measure of the feature vector resulting from the error concealment technique, on the Viterbi algorithm

$$\Phi_{j,t} = \max_i \{\Phi_{i,t-1}\, a_{ij}\} [b_j(\mathbf{o}_t)]^{\gamma_t} \quad (1)$$

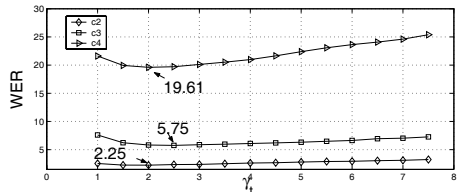

Fig. 3. *Results with covariance matrix weighting.*

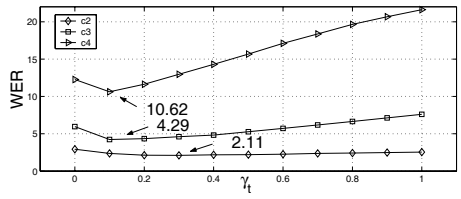

Fig. 4. *Results with weighted Viterbi decoding.*

where $\Phi_{j,t}$ is the maximum likelihood of observing feature vectors $\mathbf{o}_1$ to $\mathbf{o}_t$ and being in state j at time t, and a_{ij} is the transition probability from state i to state j.

3.2. Covariance matrix weighting

Another approach consists of weighting the Gaussian covariance matrix. Since we work with Gaussian distribution functions, this approach is related to the previous one.

In this case the decoder will consider the following probability density function with $\gamma_t \geq 1$,

$$f(\mathbf{o}) = \frac{1}{\sqrt{(2\,\pi)^d\, \gamma_t^d\, |\Sigma|}} \exp[-\frac{1}{2}(\mathbf{o}-\mu)(\gamma_t\, \Sigma)^{-1}(\mathbf{o}-\mu)^T] \quad (2)$$

where, d is the dimension of the observation vector $\mathbf{o}$, μ and Σ are the mean vector and the covariance matrix of the Gaussian, respectively.

In any method, the weighting factor can be constant or time-varying. A time-varying weighting factor will cope with the fact that the longer the burst is, the less effective the repetition concealment technique is. In the experimental section we will present results with two different laws of time variation.

4. EXPERIMENTAL RESULTS USING AURORA 2

In Table 1, some results are presented to be used as reference throughout this section. We have performed two baseline experiments for each Gilbert condition: without error mitigation, and using the repetition scheme proposed in Aurora. Our goal from now on will be to find a strategy that allows us to further improve the results of Table 1. Condition 1 will not be used, as no significant improvement of those values can be expected.

4.1. Covariance matrix weighting

The results obtained using weighted covariance matrix are shown in figure 3. Note that factor $\gamma_t = 1$ corresponds to column 2 in table 1. A simple inspection of the figure shows that there is a significant decrease of WER in comparison with the reference method. The relative improvements are 11.76%, 24.44% and 9.30% for conditions 2,3 and 4 respectively.

4.2. Weighted Viterbi decoding

As can be observed from figure 4, this strategy produces better results for all conditions than the previous one. Relative improvements with respect to the reference method are: 17.25% (0.44 absolute percent points), 43.63% (3.32 percent points) and 50.88%

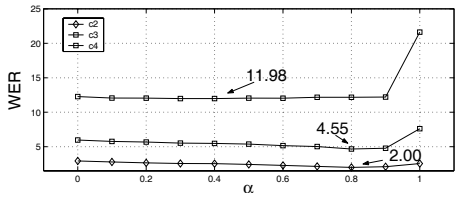

Fig. 5. *Weighted Viterbi with linearly varying factor.*

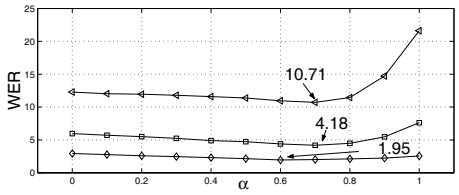

Fig. 6. *Weighted Viterbi with exponentially varying factor.*

(11.00 points) for conditions 2,3 and 4 respectively. As expected, optimal factor γ_t is found to be smaller for heavier losses. Note that in conditions 3 and 4, results are better with $\gamma_t = 0$ than with $\gamma_t = 1$, meaning that for long bursts, it is better to totally dismiss the observation probabilities than to use the NFR scheme.

4.3. Time varying weighting factor

Although this strategy may be applied both to equations 1 and 2, our experiments have proven that there is no substantial improvement if used with covariance matrix weighting, so this method will be discarded from the remainder of the paper. We propose two methods for varying γ_t: linearly and exponentially.

1. *Linear variation*

$$\gamma_t = \begin{cases} 1-(1-\alpha)\,n, & n = 1 \cdots N/2 \\ 1-(1-\alpha)\,(N-n+1), & n = N/2+1 \cdots N \end{cases} \quad (3)$$

2. *Exponential variation*

$$\gamma_t = \begin{cases} \alpha^n, & n = 1 \cdots N/2 \\ 1-\alpha^{(N-n+1)}, & n = N/2+1 \cdots N \end{cases} \quad (4)$$

where N is the lost frame gap length, n is the frame position on the gap and $0 \leq \alpha \leq 1$. Note that in the linear and exponential case $\alpha = 1$ means $\gamma_t = 1$ which leads to a hard decoding. On the other hand, $\alpha = 0$ means $\gamma_t = 0$, i.e., the probability of observation for the current frame is discarded in the Viterbi algorithm (speech frame is completely unreliable). When γ_t becomes negative in equation 3,it is floored to zero.

4.3.1. Linearly varying weighting factor

Results for this experiment are shown in figure 5. As can be observed, there is no improvement if compared with fixed weighting factor. Nevertheless, an interesting conclusion may be drawn from the graphic: best α factor is 0.8 in most conditions (for condition 4, the variation is only 0.2 percent points in the range 0.1-0.9). This may be interpreted as meaning that only the probabilities of the four first and last repetitions are reliable (γ_t is zero for n=5). A conclusion is that on average, the nearest received frame is only representative over a time period of approximately 40 ms.

4.3.2. Exponentially varying weighting factor

This method performs better than the previous one, as shown in figure 6. If compared with fixed weighting factor, an absolute decrease of 0.16 and 0.11 percent points in conditions 2 and 3, and an increase of 0.1 points in condition 4 can be observed. As observed with linearly varying weighting, there is an optimal value ($\alpha = 0.7$) for all conditions. This method may be considered equivalent to fixed weighting factor with the advantage that a single factor may be used for all conditions, without any prior knowledge of the burst length.

Cond	Danish	Finnish	Spanish	German	Average
	With no losses				
	16.65	5.17	5.47	7.51	8.70
	With no error protection				
C2	21.36	8.59	20.92	12.48	15.83
C3	25.88	12.54	24.32	17.71	20.11
C4	34.20	23.11	34.30	27.47	29.77
	With nearest frame repetition				
C2	20.16	6.92	8.58	10.32	11.49
C3	24.97	10.64	13.74	15.16	16.12
C4	35.89	20.46	26.27	26.35	27.24

Table 2. *Reference results for Aurora 3 (WER).*

5. EXPERIMENTAL RESULTS USING AURORA 3

In this section, we will validate the conclusions extracted with Aurora 2 using the noisier database Aurora 3. Table 2 shows the reference results for this section. All decoding strategies described in previous sections have been experimented with this database, but due to space restrictions, only Viterbi weighted decoding, with fixed and exponentially varying factor is shown in Table 3. Best values for each language/condition are shaded in grey for the sake of clarity.

With fixed factor (top half of the table), the improvements are higher with conditions 3 and 4 (1.8, and 6.12 percent points, 11.3% and 22.47% respectively), and almost non significant with condition 2 (0.37 points, 3.2%). The optimal factor is smaller with heavier losses, as expected. Using exponentially varying weighting factor (bottom half of the table), the dependence on the burst length is again removed. A value of $\alpha = 0.8$ is optimal for all conditions, not only on average but also for every language/condition (except for Danish, which seems to perform quite differently to the other languages). The decrease on WER is 0.56, 1.6, and 5.10 percent points (4.9%, 10.1% and 18.65%) for conditions 2, 3 and 4 respectively. Note that there is also an increase in WER, with respect to fixed factor, with conditions 3 and 4, of 0.2 and 1.1 percent points.

By way of conclusion, although the general behaviour is the same as observed with Aurora 2, there are several differences due to the noisy nature of the database, which in turn produces a high insertion rate. The differences between accuracy and word recognition rate are as high as 9.2, 3.77 and 4.65 percent points for Danish, Finnish and Spanish, while only 0.98 for German. This fact makes it difficult to extract conclusions from this study. Most improvements in WER obtained are the result of a decrease on insertion rate, probably caused by the loss of silence frames. The same, or perhaps better results might be achieved using a simple word insertion penalty.

6. CONCLUSIONS

In this paper we have explored several decoding strategies to palliate errors due to network losses in distributed speech recognition. We have shown that, if a repetition scheme as the proposed in Aurora is used, the inclusion of a weighting factor in the decoding stage may improve the results several points, particularly in a situation with heavy losses.

Fixed weighting factor											
		γ_t									
	Cond	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Danish	2	35.77	27.56	23.98	21.98	21.05	20.31	19.91	19.81	19.77	19.95	20.16
	3	40.86	30.78	26.83	25.10	23.67	23.19	23.25	23.30	23.67	24.28	24.97
	4	42.29	34.00	32.33	32.17	32.06	32.44	32.99	33.74	34.29	35.02	35.89
Finnish	2	10.34	8.40	7.60	7.12	6.79	6.55	6.61	6.65	6.65	6.79	6.92
	3	13.99	9.99	8.79	8.39	8.33	8.39	8.42	8.74	9.03	9.37	9.64
	4	17.54	13.63	13.33	13.91	14.47	15.19	16.12	17.11	18.42	19.46	20.46
German	2	10.64	8.68	8.23	8.09	8.03	8.07	8.07	8.11	8.33	8.46	8.58
	3	15.33	11.52	11.04	11.02	11.18	11.76	12.16	12.36	12.96	13.36	13.74
	4	23.66	19.64	19.56	20.46	21.22	22.00	22.78	23.74	24.56	25.65	26.27
Spanish	2	15.70	12.00	10.86	10.27	10.08	9.93	9.91	9.91	10.00	10.08	10.32
	3	22.82	15.96	13.99	13.23	13.13	13.27	13.73	14.09	14.37	14.91	15.16
	4	26.63	19.56	19.27	19.67	20.42	21.39	22.39	23.49	24.73	25.67	26.35
Average	2	18.11	14.16	12.67	11.86	11.49	11.21	11.12	11.12	11.19	11.32	11.49
	3	23.25	17.06	15.16	14.43	14.08	14.15	14.39	14.62	15.01	15.48	15.88
	4	27.53	21.71	21.12	21.55	22.04	22.75	23.57	24.52	25.50	26.45	27.24
Exponentially varying weighting factor											
		α									
	Cond	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Danish	2	35.77	33.83	31.33	29.43	27.06	24.90	22.78	20.86	19.79	19.65	20.16
	3	40.86	39.88	38.69	36.94	35.13	32.94	30.	26.67	24.47	23.26	24.97
	4	42.29	41.86	41.57	41.24	40.77	40.10	38.96	37.01	34.15	32.85	35.89
Finnish	2	10.34	9.85	9.34	8.73	8.09	7.55	7.09	6.78	6.55	6.57	6.92
	3	13.99	13.67	13.32	12.78	12.	10.63	9.81	8.96	8.31	8.44	9.64
	4	17.54	17.46	17.33	17.30	17.15	16.90	16.24	15.38	14.25	15.09	20.46
German	2	10.64	10.18	9.78	9.28	9.02	8.52	8.01	7.81	7.63	8.09	8.58
	3	15.33	14.97	14.49	13.84	13.16	12.14	11.40	10.98	11.08	12.06	13.74
	4	23.66	23.20	23.02	22.70	22.28	21.74	21.00	20.26	19.76	21.38	26.27
Spanish	2	15.70	14.85	13.80	12.97	12.14	11.26	10.46	9.98	9.74	9.66	10.32
	3	22.82	22.13	21.46	20.56	19.20	17.94	15.86	14.19	13.26	13.52	15.16
	4	26.63	26.53	26.51	26.30	25.97	25.21	24.17	22.08	20.48	21.26	26.35
Average	2	18.11	17.18	16.06	15.10	14.08	13.06	12.08	11.36	10.93	10.99	11.49
	3	23.25	22.66	21.99	21.03	19.87	18.41	16.77	15.20	14.28	14.32	15.88
	4	27.53	27.26	27.11	26.88	26.54	25.99	25.09	23.68	22.16	22.64	27.24

Table 3. *Results for Aurora 3 using Viterbi weighted decoding (WER).*

In this direction, several weighting strategies have been proposed. A fixed factor has proven effective, but the optimal value strongly depends on the length of the burst. With the use of a linearly or exponentially varying factor, this dependence is avoided. These two strategies have performed in a similar way, the last one being slightly better in most experiments.

Further work involves a more in-depth study of the results over Aurora 3 (perhaps using a VAD to reduce the insertion rate), and the extension of the experiments to Aurora 4. Other algorithms may also be experimented. A combination of several of the schemes described, like covariance matrix and observation probabilities weighting, or fixed factor followed by a linear or exponential decrease, might improve the results. More sophisticated strategies may involve the use of correlation between frames, or the computation of some kind of distance between last and first received frames.

7. REFERENCES

[1] D. Quercia, L. Docio-Fernandez, C. Garcia-Mateo, L. Farinetti, and J.C. De Martin, "Performance Analysis of Distributed Speech Recognition over IP Networks on the Aurora Database," in *Proc. ICASSP*, May 2002.

[2] L. Docio-Fernandez and C. Garcia-Mateo, "Distributed Speech Recognition over IP Networks on the Aurora 3 Database," in *Proc. ICSLP*, September 2002.

[3] "AURORA Project Database 2&3," http://www.elda.fr/proj/aurora2.html.

[4] ETSI ES 202 050 V1.1.1, "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms," Tech. Rep., ETSI, October 2002.

[5] J.C. Bolot, "End-to-end frame delay and loss behavior in the Internet," in *In Proc. ACM SIGCOMM*, Sept. 1993, pp. 289–298.

[6] A. Cardenal-Lopez, F.J. Dieguez-Tirado, and C. Garcia-Mateo, "Fast LM look-ahead for large vocabulary continuous speech recognition using perfect hashing," in *Proc. ICASSP*, May 2002, pp. 705–708.

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

[8] A. Bernard and A. Alwan, "Joint channel decoding – Viterbi recognition for wireless applications," in *Proc. EUROSPEECH*, September 2001.