Missing Feature Theory applied to Robust Speech Recognition over IP Network

Toshiki Endo[1], *Shingo Kuroiwa* [1,2], *and Satoshi Nakamura*[1]

[1]ATR Spoken Language Translation Research Labs, Kyoto, 619-0288 JAPAN
[2]University of Tokushima, Tokushima 770-8506, JAPAN
{toshiki.endo, shingo.kuroiwa, satoshi.nakamura}@atr.co.jp

Abstract

This paper addresses the problems involved in performing speech recognition over mobile and IP networks. The main problem is speech data loss caused by packet loss in the network. We present two missing-feature-based approaches that recover lost regions of speech data. These approaches are based on reconstruction of missing frames or on marginal distributions. For comparison, we also use a tacking method, which recognizes only received data. We evaluate these approaches with packet loss models, i.e., random loss and Gilbert loss models. The results show that the marginal-distributions-based approach is most effective for a packet loss environment; the degradation of word accuracy is only 5% when the packet loss rate is 30% and only 3% when mean burst loss length is 24 frames.

1. Introduction

As Internet technologies have grown, speech functions, such as the speech recognition function, have been incorporated. Moreover, with the widespread use of mobile phones and PDAs, speech recognition services over the Internet and mobile networks are expected to increase. However, low-bit rate voice codecs are not suitable for speech recognition, and so speech recognizers are likely to have poor recognition performance. Speech processing resources at the client terminal is also a significant problem.

To address these problems, the Distributed Speech Recognition (DSR) system has been standardized in the ETSI Aurora group [1][2]. In DSR, speech features are calculated and compressed at the client terminal and then transmitted over the network to the server. At the server, features are decompressed and recognition is performed.

Since low latency is expected for voice dialogues between a user and a voice service, it is desirable to use the DSR payload in an RTP-based session [3]. However, packets are discarded at the routers in the case of a congested network. Then, voice data is lost, since RTP is not a re-send mechanism for lost packets. Moreover, burst loss occurs in the actual network; therefore, degradation of speech recognition performance is considered significant [4]-[7].

There are several approaches to deal with packet loss in ASR, and the three major approaches are described bellow. The first method uses forward error correlation (FEC) for recovery of the lost data [4]. The second method is the data re-sending approach to avoid data loss [5]. However, the first method has a tradeoff relationship between speech recognition performance and compression rates, while the second method has a trade-off between recognition performance and delay. Moreover, the re-sending method causes much additional traffic load. The third method uses data interpolation to fill in the lost frames [6][7], but this may have difficulty with burst loss.

Data interpolation methods have been compared with the marginal distributed approach (marginalization method) for recognition of noise-added data [8]. In this paper, we propose also applying the marginalization method to packet loss. We present simulation results by using Internet-based DSR when data interpolation, marginalization, and tacking methods are applied. We use random loss and Gilbert loss models as the packet loss models in our simulation.

The rest of the paper is organized as follows. In sections 2, 3 and 4, we described the data interpolation method, marginalization, and the tacking method, respectively. In section 5, we describe our experimental framework. Section 5 presents the results with the random loss and Gilbert loss models and section 6 gives our conclusions.

2. Data interpolation for packet loss

The data interpolation approach is applied to the continuous density HMM (CDHMM) framework, and it uses estimated feature vectors for speech recognition. Fig. 1 illustrates the data interpolation function block at the DSR remote server. Two steps are needed for data interpolation: frame loss detection and feature vector estimation. The advantage of this method is that it needs no modification of the speech recognizer.

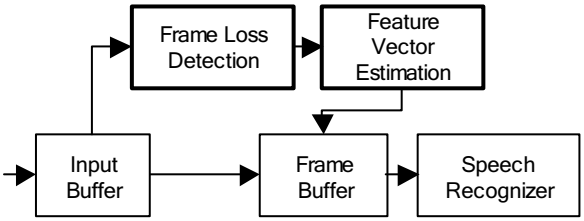

Fig. 1: Block diagram of data interpolation approach

2.1. Frame loss detection

Fame loss detection is processed in the preparation of data interpolation. In the case of RTP transmission, packet loss can be detected by reference to the sequence number in the RTP header. By using data size information in the UDP header and a sequence number in the RTP header, the number of loss frames can also be calculated, since the frame pairs for DSR data is a fixed size [1]-[3].

2.2. Feature vector estimation

Data interpolation is applied to each element of the feature vector. In this paper, we simulate two estimation methods as described below. In the following, $x = \{X_1, X_2, \cdots, X_N\}$ form a speech vector stream. In addition, it is assumed that the m -th ($1 \le m \le N$) frame is lost.

2.2.1. Data interpolation with average of study data ***(DI-ASD)***

In the first data interpolation method, lost data is interpolated by the average of the training data. Then, lost feature vector $\hat{X}_m$ is estimated as

$$\hat{X}_m = \overline{D}\,, \tag{1}$$

where $\overline{D}$ is the average of training data. We assumed this method as a baseline method.

*2.2.2. Data interpolation with received data (**DI-RD**)*

The second data interpolation method uses circumjacent data for estimation of lost data. Then, lost feature vector $\hat{X}_m$ is estimated with m in the range $f < m < b$ as follows,

$$\hat{X}_m = \frac{t_b - t_m}{t_b - t_f} X_f + \frac{t_m - t_b}{t_b - t_f} X_b\,, \tag{2}$$

where X_f and X_b are the two mean vectors on either side of the lost feature vectors, and t_f and t_b are the times of those two vectors. X_f, X_b, t_f and t_b are calculated as follows,

$$X_f = \frac{1}{N_f}\sum_{i=1}^{N_f} X_i, \quad X_b = \frac{1}{N_b}\sum_{i=1}^{N_b} X_i\,, \tag{3}$$

$$t_f = \frac{1}{N_f}\sum_{i=1}^{N_f} t_i, \quad t_b = \frac{1}{N_b}\sum_{i=1}^{N_b} t_i\,, \tag{4}$$

where N_f and N_b are the numbers of feature vectors used for data interpolation, and t_i is the time of the i-th frame.

3. Marginalization

The second method used to solve the packet loss problem is a marginalization method. A block diagram for the marginalization method is shown in Fig. 2. Received feature vectors are not processed but put into the speech recognizer directly. Adapting estimation computations requires very little extra complexity and only a modification of the existing speech recognizer. For this process, frame loss detection is also needed, and it is processed in the same manner as in the case of data interpolation.

At the speech recognizer, the marginalization method is also applied in CDHMM. The likelihood function in the HMM node C is given by

$$p(X_i \mid C) = \begin{cases} \sum_{j=1}^{M} w_j N(X_i; \mu_j, \sigma_j^2), & (i \in present) \\ Const. & (i \in missing). \end{cases} \tag{5}$$

Here, M represents the number of mixture components, w_j represents the mixture weight for mixture component j, $N(X_i; \mu_j, \sigma_j^2)$ represents a univariate Gaussian distribution function for the input feature of the i-th frame X_i, and mixture component j has a variance σ_j^2 and mean μ_j.

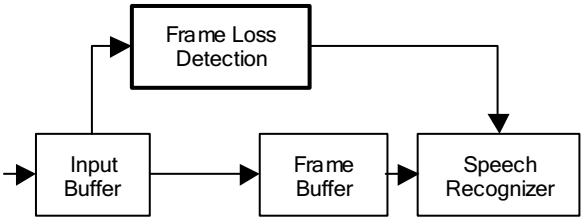

Fig. 2: Block diagram of marginalization method

4. Tacking

The tacking method is a simple recognition method, which simply treats received data as if they are continuous data. In this method, received feature vectors are simply recognized as a no-loss condition.

In this method, modifications are not needed for preprocessing of recognition and recognizer as illustrated in Fig. 3.

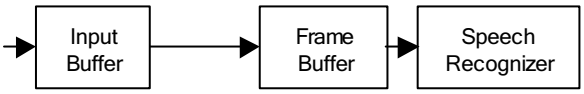

Fig. 3: Block diagram of tacking method

5. Experimental framework

5.1. Test conditions

Our experiments used the back-end framework as defined by the ETSI Aurora group [9]. We used the Aurora2 database, which is a TI digit database artificially distorted by adding noise and using a simulated channel distortion.

HTK is used for training (multi-condition) and

recognition. The digits are modeled as whole word HMMs that have 16 states with 3 Gaussians per state. The silence model has 3 states with 6 Gaussians per state. A one-state short pause model is used and tied with the second state of the silence model.

A vector size of 39 is used for recognition, which includes the cepstral coefficients (without the zero-th coefficient) and the logarithmic frame energy plus the corresponding delta and acceleration coefficients.

5.2. Network scenarios

Our experiments use two major packet loss models: the random loss and Gilbert loss models. For simplification, we assumed that each packet had one frame. Details are given below.

5.2.1. Random loss model

The random loss model is the simplest loss model, in which each packet is independent. We can simulate this model only by setting the loss probability for each frame. However, this model is much different from actual network behavior.

5.2.2. Gilbert loss model

A two-state Markovian loss model, named the Gilbert loss model [10], can have packet loss that is independent on a frame-by-frame basis like an actual network. In this model, state transition is frame-by-frame with the transition probability p and q, as illustrated in Fig. 4.

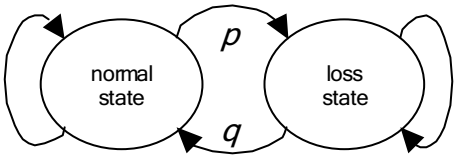

Fig. 4: Gilbert loss model

Then, mean burst loss length L_b is calculated by geometric distribution with the probability q, that is, $L_b = 1/q$. Concurrently, packet loss rate P_l is given by $P_l = p/(p+q)$. Therefore, we can simulate the desired packet loss rate and mean burst loss length only by setting the transition probability p and q.

6. Results

6.1. Random loss evaluation

We evaluated the performance of speech recognition with the random loss rate from 0% to 30%. The recognition results are shown in Fig. 5.

It was found that the word accuracy of the marginalization method, the other data interpolation method (DI-RD), and the tacking method were improved over the baseline method (DI-ASD). In particular, marginalization showed the best result, where the degradation of word accuracy was 1% while packet loss increased to 30%.

At less than 15% packet loss, the word accuracy of tacking was better than that of DI-RD. This is because the number of deletion errors is smaller than the sum of insertion errors and substitution errors. However, as packet loss increased, the number of deletion errors increased rapidly, and thus the word accuracy of the tacking method decreased.

We also investigated two types of DI-RD methods, where parameter N_b equals 0 and 1. It was found that word accuracy of the former was better than that of the latter. Although the former method is assumed capable of reconstructing the loss frame in the case of clean data, its estimation is not correct in the case of a noisy environment.

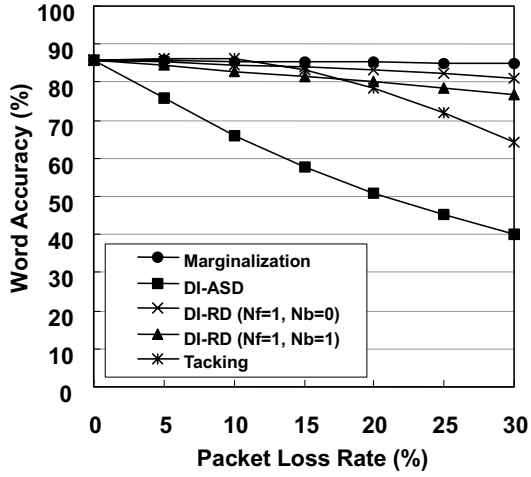

Fig. 5: Word accuracy against Packet loss rate with random loss model

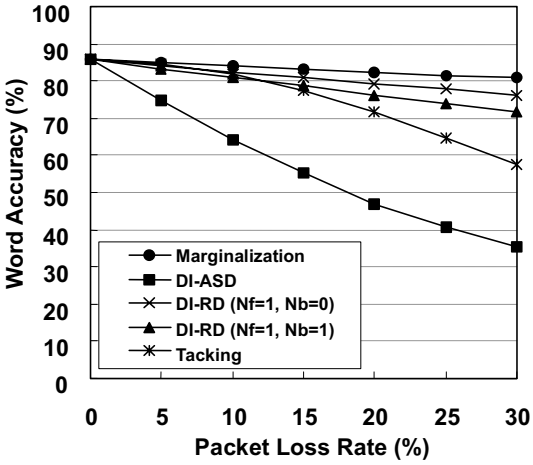

Fig. 6: Word accuracy against Packet loss rate with Gilbert loss model

6.2. Gilbert loss evaluation

Fig. 6 compares the word accuracies of the marginalization, DI-ASD, DI-RD, and tacking methods with the Gilbert loss model, where mean of burst loss length is 4. Compared with the random loss result, degradation of word accuracy against packet loss rate was large. It was also considered that marginalization was robust for burst loss, since degradation of marginalization was about 5% while packet loss was 30%.

We also evaluated word accuracy against mean burst loss length when packet loss equaled 10% as plotted in Fig. 7. As mean burst loss length increased, the word accuracy of the marginalization, DI-RD, and tacking methods decreased, while on the other hand the word accuracy of DI-ASD increased.

It was found that degradation of word accuracy with the marginalization method was about 3%, which was smaller than that of DI-RD and tacking. Consequently, marginalization is relative robust against the burst packet loss.

On the other hand, it was found that the word accuracy of DI-ASD increased as the mean burst loss length increased. This is because the sum of the substitution errors and the insertion errors decreased, since noisy data was lost.

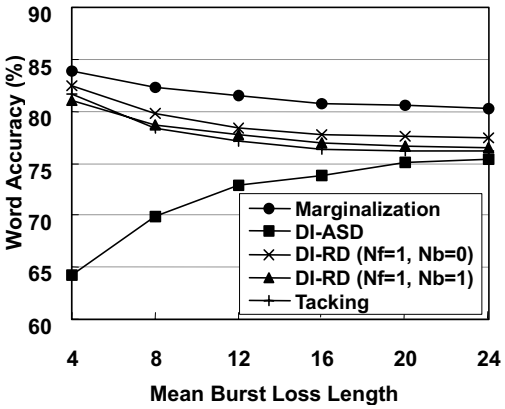

Fig. 7: Word accuracy against mean burst loss length with Gilbert loss model

7. Conclusions

We proposed applying a marginalization method for packet loss and compared different approaches to the packet loss problem with the task of DSR under the noisy condition. In our experiments with random loss and Gilbert loss models, we demonstrated that the marginalization method is more effective than other approaches to the packet loss problem in the case of a large packet loss rate and a long burst loss length. The degradation of word accuracy was only 5% when the packet loss rate was 30%, and only 3% when mean burst loss length was 24 frames.

8. Acknowledgements

The research reported here was supported in part by a contract with the Telecommunications Advancement Organization of Japan entitled, “A study of speech dialogue translation technology based on a large corpus”.

9. References

[1] ETSI Standard, “Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms”, ETSI ES 201 108 v.1.1.2, Apr. 2000.

[2] ETSI Standard, “Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms”, ETSI ES 202 050 v.1.1.1, Oct. 2002.

[3] Q. Xie, “RTP payload format for ETSI ES 201 108 Distributed speech recognition encoding”, IETF AVT-WG, draft-ietf-avt-dsr-05.txt, Oct. 2002.

[4] E. A. Riskin, et al., “Graceful degradation of speech recognition performance over lossy packet networks”, in *Proc. of Eurospeech2001*, Sept. 2001, pp. 2715-2718.

[5] D. Quercia et al., “Performance analysis of distributed speech recognition over IP networks on the aurora database”, in *Proc. of IEEE ICASSP*, May 2002, pp. 3820-3823.

[6] B. Millner and S. Semnani, “Robust speech recognition over IP networks”, in *Proc. of IEEE ICASSP*, June 2000, pp. 1791-1794.

[7] B. Millner, “Robust speech recognition in burst-like packet loss”, in *Proc. of IEEE ICASSP*, May 2001, pp. 261-264.

[8] M. Cook et al., “Robust automatic speech recognition with missing and unreliable acoustic data”, *Speech Communication 34*, 2001, pp. 267-285.

[9] H-G. Hirsch and D. Pearce, “Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions”, *ISCA ITRW ASR 2000*, Sept 2000.

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