INTERSPEECH 2009 BRIGHTON

A Media-Specific FEC Based on Huffman Coding for Distributed Speech Recognition

Young Han Lee and Hong Kook Kim

Department of Information and Communications
Gwangju Institute of Science and Technology
1 Oryong-dong, Buk-gu, Gwangju 500-712, Korea
{cpumaker, hongkook}@gist.ac.kr

10.21437/Interspeech.2009-690

Abstract

In this paper, we propose a media-specific forward error correction (FEC) method based on Huffman coding for distributed speech recognition (DSR). In order to mitigate the performance degradation of DSR in noisy channel environments, the importance of each subvector for the DSR system is first explored. As a result, the first subvector information for the mel-frequency cepstral coefficients (MFCCs) is then added as an error protection code. At the same time, Huffman coding methods are applied to compressed MFCCs to prevent the bit-rate increase by using such protection codes,. Different Huffman trees for MFCCs are designed according to the voicing class, subvector-wise, and their combinations. It is shown from the recognition experiments on the Aurora 4 large vocabulary database under several noisy channel conditions that the proposed FEC method is able to achieve the relative average word error rate (WER) reduction by 9.03~17.81% compared with the standard DSR system using no FEC methods.

Index Terms: Distributed speech recognition, forward error correction (FEC), media-specific FEC, MFCC, Huffman coding

1. Introduction

With the advancement of technology associated with wireless network systems, the demand for wireless and mobile devices has also dramatically increased. These portable devices are typically small in size and difficult to manipulate. Thus, speech recognition is a promising user interface to make these devices easier to use, since speech recognition using a microphone can take the place of a keyboard or a touch pad [1]. A major problem, however, is in that the high computational complexity of speech recognition is insufficient for most portable devices. Therefore, a new approach, distributed speech recognition (DSR), was developed to implement speech recognition in portable devices. In particular, the European Telecommunication Standards Institute (ETSI) has published several versions of DSR front-end standards, the most recent of which is defined in [2-3].

DSR operates by splitting the functions of speech recognition into a front-end and a back-end; the former is performed in the portable device and the latter in a designated speech recognition server with a high computational power. The primary purpose of the DSR front-end is to extract speech recognition features, such as the mel-frequency cepstral coefficients (MFCCs), commonly used for speech recognition. The front-end then compresses the MFCCs into the smallest possible number of bits and transmits them to the speech recognition server over a network.

One problem, however, is that when channel errors occur, DSR performance can degrade due to distortion of the MFCCs decoded at the server. There are several techniques which can assist in overcoming this problem. Forward error correction (FEC) is actually one of the typical techniques. In general, assigning more bits to an FEC will improve the quality of speech recognition [4]. Therefore, it is important to not only reduce the bits for the FEC using an unequal error protection (UEP), but also reduce the bits for MFCCs to accommodate the error protection bits. The UEP method for a DSR system has been previously proposed in [4]; however, this method was designed for channel coding.

Several coding methods for compressing MFCCs have been proposed [1-2, 6-8]. In early attempts at compression, scalar quantization or vector quantization was applied to MFCCs, and the word error rates (WERs) of a DSR system were measured according to various bit-rates [5]. To reduce the bit-rate against the quantization applied to individual frame, the interframe correlation property of MFCCs was utilized for quantization [6]. In addition, a transform coding technique using a discrete cosine transform (DCT) was proposed for MFCC compression. This technique used both the intra-frame and the inter-frame correlations of MFCCs [7]. However, these techniques increased the WER while the bit-rate was decreased.

Alternatively, the entropy coding technique, using the compressed MFCC, can achieve the bit-rate reduction with no degradation in speech recognition performance. An entropy coding technique developed for video coding was applied to MFCCs extracted under the ETSI DSR framework [8]. In this framework, speech frames were grouped using the concept referred to as a group of pictures (GOPs) followed by the application of Huffman coding [9] to each group. This reduced the bit-rate for MFCCs from 44 to 34.4 bits/frame. This technique, however, caused the error propagation from the predictive frame and the bi-predictive frame techniques.

In this paper, we first explore the importance of compressed MFCCs and the energy subvector in order to design the FEC method. We then propose a media-specific FEC method for a DSR system based on the speech recognition experiments. Following this, to reduce the bit-rate of MFCC, we measure the variance of the entropy of the feature parameters according to the voicing class. This variance implies that MFCCs and log energy in the same class have higher redundancies, allowing the bit-rates of the compressed MFCCs and log energy to be further reduced by using an entropy coding method. Therefore, we propose a class-dependent Huffman coding method to further obtain a coding gain over the compression of MFCCs and log energy according to the voicing class. In addition to such class-dependent Huffman coding, this method utilizes the principle

Table 1. *Word error rate (%) of different DSR systems on the task of the Aurora 4 database under a frame loss rate of 10%*

Test condition	Baseline (no error)	10% error	Perfectly protected single parameter (10% error)						
			(C_1,C_2)	(C_3,C_4)	(C_5,C_6)	(C_7,C_8)	(C_9,C_{10})	(C_{11},C_{12})	(C_0, log-E)
Clean	17.68	18.78	18.64	18.71	18.45	18.82	18.90	18.56	18.67
Car	19.71	21.29	20.85	21.66	26.36	21.62	21.44	21.51	21.18
Babble	25.01	26.92	36.34	27.11	26.56	26.85	26.74	26.48	27.07
Restaurant	30.72	32.56	31.31	31.71	31.90	31.82	32.19	31.90	32.08
Street	27.99	29.87	29.80	29.54	29.91	29.54	30.06	30.06	30.13
Airport	26.74	28.77	28.21	28.58	28.43	28.73	27.96	28.69	28.25
Train station	31.27	32.73	32.81	32.54	32.28	33.00	32.17	32.20	32.06
Average	25.59	27.27	26.85	27.12	26.98	27.20	27.07	27.06	27.06

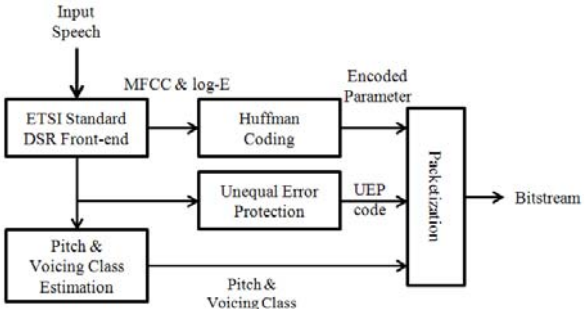

Figure 1: *Proposed media-specific FEC method based on Huffman coding for the DSR framework.*

that the correlation between MFCC subvectors within the frame is lower than the correlation of each subvector between frames. This brings us to consider Huffman coding in a subvector-wise fashion. In addition to this, we propose a hybrid Huffman coding that combines class-dependent with subvector-wise Huffman coding based on entropy comparison. Finally, we compare the average word error rate (WER) of the standard DSR system applied to the Aurora 4 database to that of the DSR system using the proposed FEC method.

The remainder of this paper is organized as follows. In Section 2, the media-specific FEC method is proposed and, in Section 3, we describe the traditional, class-dependent, and subvector-wise Huffman coding methods. We can then combine class-dependent coding with subvector-wise coding to take the advantages of each method for compressing feature parameters. In Section 4, the performance of the proposed FEC method is measured and compared with the standard DSR system. Finally, we conclude in Section 5.

2. Media-Specific FEC Method

In order to determine the protection subvector, we use the average WER to analyze the effect of each subvector on the speech recognition performance. This is a necessary investigation to design a media-specific FEC method. To this end, we selected the Aurora 4 large vocabulary database [10] which is a standard DSR database used by ETSI to evaluate the performance of large vocabulary continuous speech recognition in the DSR framework. All utterances were sampled at a rate of 16 kHz. There are two different versions for training, clean-condition training and multi-condition training, and the multi-condition training set was used here. In order to test the proposed Huffman coding methods, we used a part of the database composed of 166 utterances under seven different background noise conditions such as clean, car, babble, street traffic, airport, restaurant, and train station noise conditions.

Table 1 shows the average WERs of the DSR systems classified by nine configurations under a frame loss rate of 10%. To generate the error patterns, an error insertion process was carried out, which is based on the Gilbert-Elliot model defined in ITU-T G.191 [11]. Here, the burst error factor, γ, was set to 0.50. The first and second columns of the table show the average WERs of the baseline DSR system under no frame loss and those of the DSR system under a frame loss rate of 10%, respectively. In this paper, when a frame loss occurred, the MFCCs of the current frame were replaced with those of the previous frame. The remaining columns of the table show the average WERs in cases when only one subvector was protected under the frame loss condition. As shown in this table, the highest priority of subvectors was the subvector of (C_1,C_2). That is, it was possible to reduce the average WERs by recovering the subvector of (C_1,C_2) in the frame loss condition. We therefore propose a media-specific FEC method, where the packet of the current frame includes the subvector of (C_1,C_2) of the previous frame to reduce the WERs under the frame loss condition.

To prevent the bit-rate from being increased when the proposed FEC method is applied, we can use the Huffman coding method to assign the reduced bits to the proposed FEC method. In other words, we generate the FEC information for the *m*-th frame, *FEC*(*m*), as

$$FEC(m)=\begin{cases}2^{N-1}-1, & for \quad 2^{N-1}-1\le \Delta idx^{C_1,C_2}(m),\\ \Delta idx^{C_1,C_2}(m), & for \quad -2^{N-1}<\Delta idx^{C_1,C_2}(m)<2^{N-1}-1,\\ -2^{N-1}, & for \quad \Delta idx^{C_1,C_2}(m)\le -2^{N-1},\end{cases} \tag{1}$$

where *N* is the floored integer of available bits by the Huffman coding method, and $\Delta idx^{C_1,C_2}(m)$ is the time difference of the (C_1,C_2) subvector, defined as

$$\Delta idx^{C_1,C_2}(m)=idx^{C_1,C_2}(m)-idx^{C_1,C_2}(m-1), \tag{2}$$

where $idx^{C_1,C_2}(m)$ is the subvector index of (C_1,C_2) in the *m*-th frame. By using the proposed FEC method, the subvector of (C_1,C_2) is protected from a single frame loss. For example, if the previous frame is lost and the present frame is received correctly, the subvector of (C_1,C_2) is recovered as

$$\begin{aligned} idx^{C_1,C_2}(m-1) &= idx^{C_1,C_2}(m)-FEC(m)\\ &= idx^{C_1,C_2}(m)-\left(idx^{C_1,C_2}(m)-idx^{C_1,C_2}(m-1)\right)\\ &= idx^{C_1,C_2}(m-1). \end{aligned} \tag{3}$$

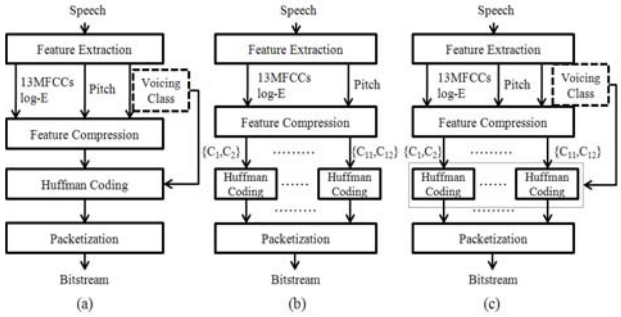

Figure 2: *Block diagrams of (a) class-dependent Huffman coding, (b) subvector-wise Huffman coding, and (c) hybrid Huffman coding.*

To recover a single frame loss, the proposed method generates an algorithm delay of about one frame length. However, it is believed that the WER is decreased because the most important subvector is correctly recovered by the proposed FEC method.

3. Huffman Coding Methods for MFCC

3.1. Traditional Huffman coding

In traditional Huffman coding, the same Huffman table is applied to each feature parameter regardless of voicing class. Two Huffman tables are generated in this paper; one for the MFCC subvectors and the other for the subvector of (C_0, log-E).

3.2. Class-dependent Huffman coding

The entropy of MFCCs and log energy varies according to the voicing class. This variability implies that MFCCs and log energy in the same voicing class have higher redundancies. Thus, the bit-rate of the compressed MFCCs and log energy can be further reduced by using a Huffman coding that is designed according to voicing class. To this end, we classify the MFCC subvectors and the subvector of (C_0, log-E) into four groups based on their voicing class. We then construct two Huffman tables for each voicing class; one for the MFCC subvectors and the other for the subvector of (C_0, log-E). Fig. 2(a) shows a block diagram of the class-dependent Huffman coding. First, the extracted MFCCs and log energy for a given analysis frame are quantized as described in Section 2. Then, the MFCC subvector indices and the subvector index of (C_0, log-E) are further compressed using the Huffman tables corresponding to the voicing class of the frame.

3.3. Subvector-wise Huffman coding

In addition to the entropy dependency on the voicing class, the entropy of the MFCCs and log energy varies according to the subvector. Similar to the interpretation previously described in Section 3.2, this variability implies that the MFCCs and log energy also have higher redundancies compared to a traditional Huffman coding, thus the bit-rate of the compressed MFCCs and log energy can be further reduced by using Huffman trees designed subvector-wise. Fig. 2(b) presents a block diagram of the subvector-wise Huffman coding, where the extracted MFCCs and log energy are quantized as described in Section 2. Each subvector is then further compressed using the Huffman tables corresponding to the subvector of the frame.

Table 2. *Average entropy comparison (bits/frame) of each subvector for the different Huffman coding methods.*

Method		C_1–C_{12}	(C_0,log-E)
Traditional Huffman coding		5.79	7.07
Class-dependent Huffman coding	Non-speech	5.50	6.55
	Unvoiced	5.73	6.85
	Mixed-voiced	5.66	6.76
	(Fully) Voiced	5.85	6.66
	Average	5.75	6.72
Subvector-wise Huffman coding	(C_1,C_2)	5.42	7.07
	(C_3,C_4)	5.02	
	(C_5,C_6)	5.48	
	(C_7,C_8)	5.32	
	(C_9,C_{10})	4.91	
	(C_{11},C_{12})	5.20	
	Average	5.24	7.07

Table 3. *Average entropy comparison (bits/frame) of the hybrid Huffman coding method.*

Method	Non-speech	Un-voiced	Mixed-voiced	(Fully) Voiced	Average
(C_1,C_2)	4.46	5.28	5.22	5.19	5.14
(C_3,C_4)	3.49	4.66	4.22	5.41	4.84
(C_5,C_6)	4.54	5.22	4.98	5.70	5.35
(C_7,C_8)	4.60	5.13	4.85	5.71	5.32
(C_9,C_{10})	4.17	4.59	4.39	5.25	4.83
(C_{11},C_{12})	4.53	4.91	4.75	5.50	5.13
(C_0,log-E)	6.55	6.85	6.76	6.66	6.72
Total	32.34	36.64	35.18	39.42	37.32

3.4. Entropy comparison

To investigate the extent to which the bit-rate of the compressed subvector indices can be reduced using Huffman coding, we evaluated the entropy of feature subvector indices according to traditional Huffman coding, class-dependent Huffman coding, and subvector-wise Huffman coding. Table 2 shows the measured entropy for each Huffman coding method. For this experiment, we used the Aurora 4 large vocabulary database [10].

As shown in the first row of Table 2, traditional Huffman coding required 5.79 and 7.07 bits/frame for the MFCC subvectors, C_1-C_{12}, and the subvector of C_0 and log-E, (C_0, log-E), respectively. On the contrary, 5.75 and 6.72 bits/frame were required for C_1-C_{12} and the subvector of (C_0, log-E), respectively, for class-dependent Huffman coding. The classification percentages for non-speech, unvoiced speech, mixed-voiced speech, and fully voiced speech were measured at 10.82, 33.78, 9.20, and 46.18%, respectively. These percentages were employed to calculate the weighted average bits/frame of class-dependent Huffman coding. In the case of subvector-wise Huffman coding, 5.24 and 7.07 bits/frame were required for C_1-C_{12} and the subvector of (C_0, log-E).

3.5. Hybrid Huffman coding

As can be seen in Table 2, the MFCC subvectors had smaller entropy when differential Huffman coding was applied. Based on this observation, it can be concluded that hybrid types of Huffman coding can be made by combining class-dependent or subvector-wise Huffman coding methods with the differential Huffman coding method. Fig. 2(c) shows the

Table 4. *Word error rate (%) of different DSR systems on the task of the Aurora 4 database under frame loss rates of 5, 10, 20, and 50%.*

Test condition	Baseline	5% error		10% error		20% error		50% error	
		Standard	Proposed	Standard	Proposed	Standard	Proposed	Standard	Proposed
Clean	17.68	18.27	17.94	18.78	18.64	20.15	19.78	26.56	25.71
Car	19.71	20.52	20.07	21.29	21.22	23.50	22.65	30.83	29.28
Babble	25.01	26.22	26.11	26.92	26.19	27.15	27.18	36.24	34.99
Restaurant	30.72	31.05	31.34	32.56	31.90	34.11	32.93	40.37	39.63
Street	27.99	29.24	28.84	29.87	30.17	31.09	30.94	39.63	38.64
Airport	26.74	26.96	26.92	28.77	28.66	30.98	30.13	37.79	36.13
Train station	31.27	31.98	32.10	32.73	32.47	34.00	34.07	41.33	41.77
Average	25.59	26.32	26.19	27.27	27.04	28.72	28.24	36.11	35.16

proposed hybrid Huffman coding method according to voicing class and subvector-wise, respectively. In other words, the differential coding is applied to both the class-dependent Huffman coding and the subvector-wise Huffman coding.

Table 3 shows the entropy comparison of the hybrid Huffman coding method. It is shown from the table that the proposed hybrid Huffman coding has the smallest entropy of the Huffman coding methods.

4. Performance Evaluation

From the experiment described in Section 3.5, the hybrid Huffman coding method can give an average bit reduction of about 6.35 bits/frame. That is, the $FEC(m)$ can use 64 indices as the time difference of the (C_1,C_2) subvector for error protection.

To evaluate the performance, the average WERs of the baseline system, the standard DSR system, and the DSR system using the proposed FEC method were compared under four different frame loss rates of 5%, 10%, 20% and 50% using the Aurora 4 large vocabulary database. Every error pattern was generated by an error insertion device using the Gilbert-Elliot model defined in ITU-T G.191 [11]. A part of the database was used for testing ASR performance, which was composed of 166 utterances recorded under seven different background noise conditions such as clean, car, babble, street traffic, airport, restaurant, and train station noise conditions.

Table 4 shows the average WER of the baseline DSR system and the DSR system employing proposed FEC method under different frame loss rates. It was shown from the table that the relative reductions of the proposed FEC method were 17.81, 13.70, 15.34, and 9.03% under frame loss rates of 5, 10, 20 and 50%, respectively. [1]

5. Conclusion

In this paper, we proposed a media-specific forward error correction (FEC) based on Huffman coding for distributed speech recognition (DSR). To reduce the word error rate (WER) in noisy channels, we investigated the importance of each subvector for the DSR system. The media-specific FEC method based on the investigation was then designed. Moreover, to prevent the bit-rate from increasing under the proposed FEC method, a hybrid Huffman coding method taking into account voicing class and subvector-wise were also applied to the compressed MFCC feature parameters. It was shown from the ASR experiments conducted on the Aurora 4 large vocabulary database that the proposed method provided a 9.03~17.81% reduction in the WER as compared with the standard DSR system.

[1] The relative reductions are calculated as

$$relative\ reduction = \frac{WER_{proposed} - WER_{baseline}}{WER_{standard} - WER_{baseline}}.$$

6. Acknowledgements

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MEST) (No. 2009-0057194), and by the Ministry of Knowledge Economy under the Information Technology Research Center support program, supervised by the Institute of Information Technology Advancement (IITA-2009-C1090-0902-0017).

7. References

[1] N. Srinivasamurthy, A. Ortega, and S. Narayanan, "Efficient scalable encoding for distributed speech recognition," Speech Communication, 48(8):888–902, Aug. 2006.

[2] ETSI ES 202 211, Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Extended Front-end Feature Extraction Algorithm; Compression Algorithms; Back-end Speech Reconstruction Algorithm, Nov. 2003.

[3] A. Sorin, et al., "The ETSI extended distributed speech recognition (DSR) standards: Client side processing and tonal language recognition evaluation," in Proc. of ICASSP, 129–132, May 2004.

[4] V. Weerackody, W. Reichl, and A. Potamianos, "An error-protected speech recognition system for wireless communications," IEEE Trans. Wireless Communications, 1(2):282-291, Apr. 2002.

[5] V. V. Digalakis, L. G. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the World Wide Web," IEEE Journal on Selected Areas in Communications, 17(1):82–90, Jan. 1999.

[6] G. N. Ramaswamy and P. S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in Proc. of ICASSP, 2:977–980, May 1998.

[7] Q. Zhu and A. Alwan, "An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition," in Proc. of ICASSP, 113–116, May 2001.

[8] B. J. Borgstrom and A. Alwan, "A packetization and variable bitrate interframe compression scheme for vector quantizer-based distributed speech recognition," in Proc. of Interspeech, 578–581, Aug. 2007.

[9] D. A. Huffman, "A method for the construction of minimum-redundancy codes," Proceedings of the IRE, 9(40):1098–1101, Sept. 1952.

[10] G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task," ETSI STQ Aurora DSR Working Group, Dec. 2002.

[11] ITU-T Recommendation G.191, Software Tools for Speech and Audio Coding Standardization, Aug. 2005.