QUALCOMM-ICSI-OGI FEATURES FOR ASR

Andre Adami[a], *Lukas Burget*[a], *Stephane Dupont*[b,0], *Hari Garudadri*[c],
Frantisek Grezl[a],*Hynek Hermansky*[a,b], *Pratibha Jain*[a], *Sachin Kajarekar*[a],
Nelson Morgan[b], *Sunil Sivadas*[a] *

[a]OGI School of Science & Engineering, OHSU, Portland, Oregon, USA.
[b]International Computer Science Institute, Berkeley, California, USA.
[c]Qualcomm Inc., San Diego, California, USA.
adami,lukas,franta,hynek,pratibha,sachin,sunil@ece.ogi.edu
dupont,morgan@icsi.berkeley.edu hgarudad@qualcomm.com

ABSTRACT

Our feature extraction module for the Aurora task is based on a combination of a conventional noise supression technique (Wiener filtering) with our temporal processing technigues (linear discriminant RASTA filtering and nonlinear TempoRAl Pattern (TRAP) classifier). We observe better than 58% relative error improvement on the prescribed Aurora Digit Task, a performance level that is somewhat better than the new ETSI Advanced Feature standard. Furthermore, to test generalization of our approach to an independent test set not available during development, we evaluate performance on American English SpeechDatCar digits and show 10.54% relative improvement over the new ETSI standard.

1. INTRODUCTION

The European Telecommunication Standards Institute (ETSI) initiated the standardisation of an advanced front-end for DSR, under the name "Aurora" [1]. Evaluation comprises of connected digit recognition tasks under a range of noise conditions on six languages. The challenge is to design a front-end that gives a significant reduction in Word Error Rate (WER) compared to the MFCC standard [3] within limited computational resources and a restricted algorithmic delay.

In this paper, we propose a robust front-end based on spectral and temporal processing. In the terminal, robust cepstral features are computed using a modified Wiener filter followed by temporal filtering. A Multi-Layer Perceptron (MLP) based Voice Activity Detector (VAD) is used to detect the non-speech frames. Features are compressed using the split Vector Quantization (VQ) algorithm and transmitted at a datarate of 4800 bps. At the server, two feature streams are generated. The first consists of the decompressed cepstral features, and the second consists of TRAPS [2] based features. The cepstral features are mean and variance normalized and concatenated with TRAPS based features. We compare the performance of the proposed algorithm with the ETSI adopted advanced front-end standard [4]. Figure 1 shows the block diagram of the proposed front-end. The following sections describe the feature computation blocks.

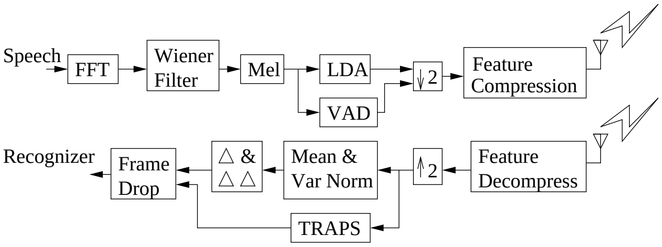

Fig. 1. Robust feature extraction for DSR

2. TERMINAL FEATURE EXTRACTION

The speech signal is sampled at 8kHz, segmented into frames of 25 ms with a shift of 10ms. A Hamming window is applied to each frame and 256 point FFT is computed. Only the first 129 points are retained after computing the short term power spectra.

2.1. Noise Compensation

To improve the Signal to Noise Ratio (SNR) of the signal, a modified Wiener filter algorithm is applied to the power spectra. Here the noise is assumed to be additive and uncorrelated. An estimate of the noise power spectrum is obtained using a first order recursion and frame energy based update. An instantaneous Wiener filter is obtained every frame that incorporates noise overestimation factor. The noise overestimation factor is a function of the local $aposteriori$ SNR.

[0]Now with Multitel, Belgium. *Authors appear in alphabetical order.

The instantaneous filter estimate is smoothed in time and frequency to reduce variance due to erroneus noise spectral estimates. Since the recursion introduces a group delay to the filter estimate, the smoothed filter is not synchonized with the instantaneous filter. We estimate this asynchrony to be 2 frames. The clean speech power spectral estimate is obtained by multiplying the smoothed filter and noisy speech power spectrum.

The clean power spectral estimate is weighted by 23 triangular weighting functions simulating Mel-scale spaced filter banks. A natural logarithm is applied to the outputs of the Mel filterbank.

2.2. Temporal filtering using RASTA-LDA filters

The filter coefficients are derived using the LDA technique on the phonetically labeled OGI-Stories database as described in [5]. Car noise at 10 dB SNR is artificially added to the database. The noisy speech files are cleaned using the noise compensation technique mentioned in Section 2.1. A 101 point feature vector, centered and labeled by one of the forty one phoneme classes at the current frame, from the 7th mel band is used for LDA. Each phoneme is divided uniformly into three states and each state is used as a class, resulting in a total of 123 classes. The leading discriminant vector from the 7th band is used as a temporal RASTA filter. To reduce the latency, the filter is truncated to a 51 point causal filter. Finally, it is convolved with a 25 Hz low-pass filter and further truncated to 30 points. Figure 2 shows the impulse and magnitude response the LDA filter. The filter is applied to the log Mel-filterbank outputs and fifteen cepstral coefficients are calculated using a Discrete Cosine Transform (DCT).

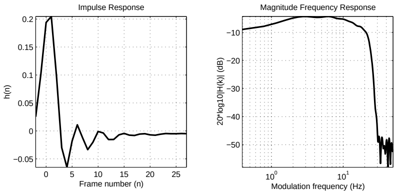

Fig. 2. RASTA-LDA filter and its magnitude frequency response

2.3. Voice Activity Detector (VAD)

The VAD is a single hidden-layer feed-forward MLP. It is trained to discriminate between speech and non-speech frames using the backpropagation algorithm. Training is done offline using a noisy database. The MLP uses 9 frames of 6 ceptral coefficients computed from low-pass filtered log-energies of the 23 Mel filters. The output of the trained MLP gives an estimate of the posterior probability of the current frame being speech or non-speech.

3. FEATURE COMPRESSION AND DECOMPRESSION

Fifteen cepstral coefficients and the silence probability are concatenated to form a single feature vector. The feature vector is downsampled by two in time. The final feature vector is compressed using a split Vector Quantization (VQ) algorithm. The LBG algorithm is used for training of the codebook. The codebook is initialized with the mean value of the entire training data. At each step, each centroid is split into two and their values are re-estimated. Splitting is performed in the positive and negative direction of the standard deviation vector multiplied by 0.2. To determine the index, the closest VQ centroid is found using the Euclidean distance. The number of bits required for the description of one frame after packing indices to the bit stream is 76. These 76 bits are protected with 16 bits of CRC. This CRC together with indices creates a frame packet of 11.5 octets. Refer to [3] for further details on bit stream formatting.

At the server, the received bitstream is decompressed to regenerate the speech feature vectors. Synchronization sequence detection, header decoding and feature decompression are implemented as in [3]. To protect the transmitted data against channel errors, Fire's correction code is used. This code is able to repair error bursts of up to five bits and to detect six bits long error bursts.

4. SERVER FEATURE GENERATION

Two streams of features are generated from the decompressed features. The first stream consists of cepstral coefficients that are upsampled, mean and variance normalized and augmented with first and second delta coefficients. The second stream consists of TRAPS based features. In [2], it was shown that augmenting cepstral features with TRAPS based features, significantly improves robustness. A threshold of 0.5 is applied to the silence probability to convert it to a binary VAD flag. The following sections explain these blocks further.

4.1. Mean and Variance Normalization

In order to compensate for the shifts in mean and variance caused by the additive and convolutive noise, a recursive mean and variance normalization is applied to the cepstral coefficients. The mean and variance are initialized using the global mean and variance estimated on a noisy speech database.The estimates of the local mean and variance are updated for each frame marked as speech by the binary VAD flag. The forgetting factor of the recursion is set to 0.01, corresponding to roughly a 1 sec time constant.

4.2. TRAPS based features

TRAPs features are based on multi-band and multi-stream approaches. For each mel-band, a feed-forward multilayer perceptron is trained to classify speech frames. They have one hidden layer with a sigmoid activation function and they are trained using a softmax activation function at their outputs. The input to each classifier is a temporal trajectory of mel spectral energies. Fifteen mel spectral values are reconstructed from the fifteen decompressed cepstral based features using the Inverse DCT (IDCT) at the server-end. Each temporal trajectory covers a context of 9 frames in the past and 9 frames in the future. The temporal trajectories are mean subtracted and variance normalized. The mean and variance are reestimated for each frame from the 19 points long temporal trajectory. The dimensionality of the normalized temporal trajectory is reduced from 19 to 10 using Principal Component Analysis (PCA). The number of hidden units for each band classifier is 25. The output units of each classifer are manner-based articulatory-acoustic categories (Vowel, Flap, Stop, Fricative, Nasal) as well as silence. The linear outputs (6 outputs before the final softmax non-linearity) from each band-classifier are concatenated to obtain a 90 dimensional vector. The dimensionality is reduced to 60 using PCA and used as input to a "merging" feed-forward MLP. The merging MLP is trained to classify the same six manner of articulation targets. The merging MLP has 200 hidden units. The band classifier MLPs, merger MLP and PCA matrix are trained using TIMIT database added with noise. The manner classes are derived from the phoneme classes by a canonical mapping.

4.3. Final feature vector

The first and second delta features are derived from the normalized cepstral features and appended to the static features. They are concatenated with the six features from the TRAPS stream. The final feature vector size is 51. The VAD flag is smoothed using a median filter with a length of twenty one points. The frames marked as silence by the VAD flag after smoothing are not sent to the back-end recognizer.

5. EXPERIMENTS AND RESULTS

As part of the advanced front-end standardization, ETSI defined development databases that contain digit strings in six different languages. They are Aurora-2 (English) and Aurora-3 (Italian, Finnish, Spanish, German and Danish). For the work reported here, we created a test set consisting of data that was not used during training or system development. The new set comprised the connected digits recording of SpeechDatCar (SDC) US English [6]. The connected digits subset of SDC-US consists of 1 sheet number (5 + digits), 1 spontaneous telephone number, 3 read telephone numbers, 1 credit card number (14-16 digits), 1 PIN code (6 digits). The database contains 7658 recordings spoken by more than 150 speakers. Recordings from close-talking and far microphone are selected. The files are down-sampled from 16 to 8 kHz followed by DC offset removal using ITU-T software tools library. The speaker synchronization beeps in the beginning of the files are cut off using an automated procedure. As in Aurora-3, three train/test configurations are defined: the well-matched condition (WM), the medium mismatched (MM) condition and the highly mismatched condition (HM). In the WM case, 70% of the entire data is used for training and 30% for testing and training set contains all the variability that appear in the test set. In the MM case, only far microphone data is used for both training and testing. For the HM case, training data consists of close microphone recordings only while testing is done on far microphone data.

5.1. Results

The recognizer is the ETSI-specified HTK-based whole word HMM system with 16 states per HMM and 3 components per state with diagonal covariance matrices. Table 1 and 2 present the performance of the proposed front-end (QIO) compared to the MFCC front-end with end-point detection [7] for the Aurora databases. The *relative improvement* for WM, MM and HM conditions are weighted by *0.40, 0.35 and 0.25* respectively. We achieve a WER reduction of more than 58% relative to the MFCC system. In Table 3 we compare the performances of QIO and ETSI advanced front-end standard relative to the MFCC system for Aurora databases. It also shows the performance of the proposed front-end without TRAPS (QIO-NoTRAPS). Table 4 presents the WER (baseline MFCC results are not available for this database) of the two front-ends and relative improvement of QIO features relative to ETSI advanced front-end standard. From the results it is evident that the proposed front-end is better than the ETSI advanced front-end standard for both the Aurora and new test sets. Finally we present, in Tables 5 and 6, the performance relative to the baseline provided for ICSLP submission. To permit replication of our results and application to other test sets, we have made the proposed front-end available to interested people at the OGI website. [1]

[1] ftp://ftp.asp.ogi.edu/pub/download/Aurora/qio-afe.tgz

5.2. System Complexity and Latency

The overall algorithmic latency of the system is 185 ms, where the terminal latency is 55 ms and the server latency is 130 ms and requires approximately 6 kWords of memory. The computational load is approximately 10 MOPS for the terminal and 2 MOPS for the server.

6. CONCLUSIONS

For the official Aurora advanced front-end evaluation, we submitted a scaled down (no TRAPS) version of our proposed system, corresponding to QIO-NoTRAPS in Table 3. Here we have described methods and results for the more complete system. We presented a feature extraction scheme that incorporates noise robustness using Wiener filtering, temporal discriminants using data-derived LDA filters, mean and variance normalization and TRAPS based features. A voice activity detector is used to discard the frames that are unlikely to contain speech. Our results show a WER reduction of better than 58% relative to the MFCC baseline on the Aurora databases. We show an improvement over the new ETSI advanced front-end standard on Aurora and SDC US databases. The proposed front-end has a low latency and complexity. One of the key potential of the proposed system resides in the multiband structure of the TRAPS approach. We expect this alternate feature stream to increase the robustness whenever the Wiener filter lacks reliable estimates of the noise power spectrum. This can happen on long portions of continuous speech during which the noise level can change relatively quickly, a condition for which state-of-the art noise spectrum estimation algorithms do not perform well.

7. ACKNOWLEDGEMENTS

The research was supported by DARPA under R16007-01 and by an industrial grant from Qualcomm Inc.

8. REFERENCES

[1] D Pearce, ”Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends”, *AVIOS2000,* San Jose, CA, May 2000.

[2] H. Hermansky, S. Sharma and P. Jain, “Data-derived nonlinear mapping for feature extraction in HMM”, *Proc. ASRU’99,* Keystone, December 1999.

[3] *ETSI ES 201 108 v1.1.2 Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms,* April 2000.

[4] *ETSI ES 202 050 Ver. 0.1.1 Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced feature extraction algorithm,* April 2002.

[5] S. S. Kajarekar and H. Hermansky, “Analysis of information in speech and its application in speech recognition”, *TSD’2000,* Brno, Czech Republic, September 2000.

[6] Peter Heeman and David Cole, “SpeechDatCar: US English”, *Oregon Graduate Institute, Version 1.2,* June 2001.

[7] *Small vocabulary evaluations: Baseline Mel-Cepstrum Performances with Speech Endpoints,* STQ Aurora DSR Working Group, Motorola, Version 1.0, June 2001.

Aurora 2 Relative Improvement (MFCC Standard)				
	Set A	Set B	Set C	Overall
Multi	32.79%	45.93%	44.36%	40.36%
Clean	69.60%	74.67%	69.56%	71.62%
Average	51.20%	60.30%	56.96%	**55.99%**

Table 1: Aurora 2 Relative improvement for QIO.

Aurora 3 Relative Improvement (MFCC Standard)						
	ITA	FIN	SPA	GER	DAN	Average
WM	59.79	60.38	64.89	40.56	56.54	55.97%
MM	69.14	51.42	67.38	42.88	48.30	55.05%
HM	68.24	77.60	78.24	55.87	69.35	71.45%
Overall	64.13	61.55	69.10	45.20	59.52	**59.52%**

Table 2: Aurora 3 Relative improvement for QIO.

	QIO-NoTRAPS	QIO	ETSI
Aurora-2(x40%)	49.84%	55.99%	54.73%
Aurora-3(x60%)	56.62%	59.52%	56.61%
Overall	**53.91%**	**58.11%**	**55.85%**

Table 3: Relative improvement for QIO and ETSI front-ends for Aurora databases.

	QIO (**WER**)	ETSI (**WER**)	Improvement
WM	3.80%	4.44%	14.41%
MM	10.51%	11.42%	7.97%
HM	10.18%	11.06%	7.96%
Overall	7.74%	8.54%	**10.54%**

Table 4: Comparison of QIO and ETSI front-end for SDC US database

Aurora 2 Relative Improvement (ICSLP Baseline)				
	Set A	Set B	Set C	Overall
Multi	30.04%	40.34%	38.68%	35.89%
Clean	68.40%	74.51%	65.44%	70.25%
Average	49.22%	57.42%	52.06%	**53.07%**

Table 5: Aurora 2 relative improvement for QIO.

Aurora 3 Relative Improvement (ICSLP Baseline)					
	FIN	SPA	GER	DAN	Average
WM	56.34	62.75	39.20	52.99	52.82%
MM	46.64	67.94	38.24	43.76	49.15%
HM	75.75	78.58	55.35	67.14	69.21%
Overall	57.80	68.52	42.90	53.30	**55.63%**

Table 6: Aurora 3 Relative improvement for QIO.