ON USING HETEROGENEOUS DATA FOR VEHICLE-BASED SPEECH RECOGNITION: A DNN-BASED APPROACH

Xue Feng[1], *Brigitte Richardson*[2], *Scott Amman*[2], *James Glass*[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA, 02139
[2]Ford Motor Company

xfeng@mit.edu, {bricha46, samman}@ford.com, glass@mit.edu

ABSTRACT

Most automatic speech recognition (ASR) systems incorporate a single source of information about their input, namely, features and transformations derived from the speech signal. However, in many applications, e.g., vehicle-based speech recognition, sensor data and environmental information are often available to complement audio information. In this paper, we show how these data can be used to improve hybrid DNN-HMM ASR systems for a vehicle-based speech recognition task. Feature fusion is accomplished by augmenting acoustic features with additional side information before being presented to the DNN acoustic model. The additional features are extracted from the vehicle speed, HVAC status, windshield wiper status, and vehicle type. This supplementary information improves the DNNs ability to discriminate phonetic events in an environment-aware way without having to make any modification to the DNN training algorithms. Experimental results show that heterogeneous data are effective irrespective of whether cross-entropy or sequence training is used. For CE training, a WER reduction of 6.3% is obtained, while sequential training reduces it by 5.5%.

Index Terms— Noise Robustness, Deep Neural Network, Additional Feature for ASR, Condition-aware DNN

1. INTRODUCTION

Recently, there is a growing demand for distant speech input for a variety of consumer products [1, 2], such as phone calls, music and navigation while driving, robotic communication systems, voice controls of mobile phones, or other portable devices. However, the noise and reverberations in these environments can dramatically degrade the performance of even state-of-the-art ASR systems.

To improve ASR robustness, different approaches have been investigated: Front-end methods include speech signal pre-processing [3, 4, 5], robust acoustic features [6, 7]; back-end methods include model compensation or adaptation [8], and uncertainty decoding [9] etc.

Recently, deep neural networks (DNNs) have become a competitive alternative to Gaussian Mixture Models (GMMs) [10]. Many researchers have also reported different ways of using DNNs to generate robust speech features. For example, [7] investigated the effectiveness of DNNs for detecting articulatory features, which, combined with MFCC features, were used for robust ASR tasks. Noise-aware training (NAT) was proposed in [11] to improve noise robustness of DNN-based ASR systems. It uses a crude estimate of noise obtained by averaging the first and the last few frames of each utterance as input to the DNN acoustic model. Similarly, [12] uses speech separation to obtain a more accurate estimate of noise. In the above prior work, the additional features are generally derived from the speech signals, and there are limited studies on utilizing existing environmental information.

In this paper, we investigate the benefit of using features extracted from available heterogeneous data in addition to the features derived from speech signals. In many ASR tasks, e.g., vehicle-based speech recognition and robotic communication systems, various data from the motor sensors, devices and camera sensor etc., are available, which may provide additional clues for ASR. We propose a DNN-based method to incorporate such information by augmenting the input speech features with additional features extracted from the heterogeneous data.

We evaluate this approach using a vehicle-based speech corpus consisting of 30 hours of data recorded from 113 speakers under a variety of conditions. The heterogeneous data include vehicle speed, vehicle model, windshield wiper and fan status, etc., which can be obtained in real time without human supervision. We demonstrate that our proposed approach successfully integrates these discrete and continuous data, and that these data are helpful in improving the robustness of the ASR system. Our experiments show that the additional features can lead up to 6.3% WER reduction in cross entropy training and 5.5% in sequence training, compared to the baseline DNN-HMM hybrid systems.

The remainder of the paper is organized as follows. Section 2 describes our proposed Heterogeneous DNN system. Section 3 introduces the dataset. Section 4 provides the experiments and results. Finally conclusions are drawn in Section 5.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

2. SYSTEM DESCRIPTION

A DNN is a multi-layer perceptron with many hidden layers between its inputs and outputs. In a modern DNN hidden Markov model (HMM) hybrid system, the DNN is trained to provide posterior probability estimates for the HMM states. Starting with a visible input **x**, each hidden layer models the posterior probabilities of a set of binary hidden variables h given the input visible variables v, while the output layer models the class posterior probabilities.

The networks are trained by optimizing a given training objective function using the error back-propagation procedure. For the DNNs in this work, we use two different loss objectives. One is Cross Entropy (CE), which minimizes frame error. The other one is sequence-discriminative training using state-level minimum Bayes risk (sMBR) criterion, which minimizes expected sentence error.

2.1. Speech Features

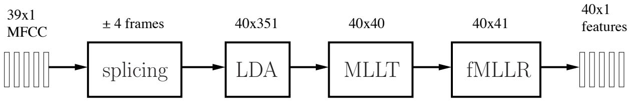

Fig. 1. *Generation of baseline speech features.*

As shown in Figure 1, the speech waveforms are first parameterized into a conventional sequence of 39-dimensional Mel-frequency cepstral coefficient (MFCC) vectors based on a 25ms Hamming window, and computed every 10ms. Cepstral mean subtraction is applied on a per speaker basis. The MFCCs are then spliced across 9 frames to produce 351 dimensional vectors. Linear discriminant analysis (LDA) is used to reduce the dimensionality down to 40 by using context-dependent HMM states as classes for LDA estimation. A maximum likelihood linear transform (MLLT) [13] is then applied to the MFCC-LDA features to better orthogonalize the data. Finally, global fMLLR [14] is applied to normalize inter-speaker variability. In our experiments fMLLR is applied both during training and test, which is known as speaker-adaptive training (SAT) [15].

2.2. Heterogeneous Features

The heterogeneous data we explore include speed, HVAC fan status, wiper status, and vehicle type. These data can be automatically reported by the vehicle in real time in parallel with the audio and require no human supervision. Table 1 lists the values of the additional data used in our experiments.

In order to fit this information into the DNN model, pre-processing are conducted on these data. We suggest that all real-valued feature vectors be normalized globally to zero mean, unit variance, while all status feature vectors be mapped to binaries of 0/1. Therefore, speed is normalized globally to zero mean, unit variance. AC fan status and wiper status are mapped to binary values. Vehicle types are mapped to five distinct values according to the size of the vehicle model, and further normalized globally to zero mean, unit variance.

Data	Type	Values
speed	real-valued	0 MPH, 35 MPH, 65 MPH
ac_fan	status	On/Off
wiper	status	On/Off
vehicle_type	categorical	model$_1$, model$_2$, .., model$_5$

Table 1. *List of available heterogeneous data*

These extracted additional features are concatenated with the spliced corresponding speech features. We append individual, combinations of two, three and all four additional features to the speech features to test the effectiveness of these different additional features.

2.3. DNN with Heterogeneous Data

For the DNN with heterogeneous data, the input speech features (fMLLR-adapted MFCC-LDA-MLLT) are computed as in the baseline system. However, these features are now augmented with various combinations of additional features computed from the heterogeneous data. The features are augmented for both training and decoding.

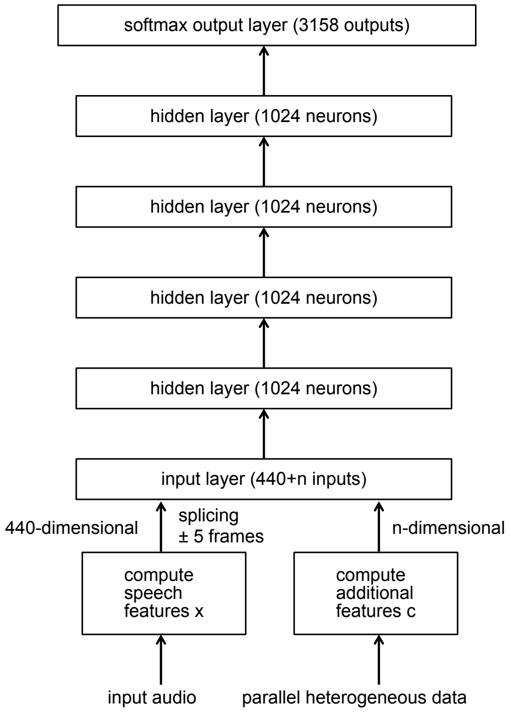
softmax output layer (3158 outputs)
hidden layer (1024 neurons)
hidden layer (1024 neurons)
hidden layer (1024 neurons)
hidden layer (1024 neurons)
input layer (440+n inputs)
440-dimensional
splicing
± 5 frames
n-dimensional
compute
speech
features x
compute
additional
features c
input audio
parallel heterogeneous data

Fig. 2. *Diagram of the proposed Heterogeneous DNN system.*

Figure 2 gives a diagram of our DNN framework that incorporates additional features. The speech features are derived from speech signals, while the additional features are provided by various sources of sensor information, e.g., camera sensor, motion sensor, speed sensor, fan power etc. As discussed in Section 2.2, in our Ford experiment, these additional features include vehicle speed, HVAC fan status, windshield wiper status, and vehicle type. Thus, the DNN's input is a super vector with the additional features appended to the speech features. At time t, the input

$$\mathbf{v}_{0t} = [x_{t-\tau}, ..., x_{t-1}, x_t, x_{t+1}, ..., x_{t+\tau}, c_t]. \qquad (1)$$

Each observation is propagated forward through the network, starting with the lowest layer v_0 . The output variables of each layer become the input variables of the next layer. In the final layer, the class posterior probabilities are computed using a softmax layer. Specifically, instead of 440 MFCC-LDA-MLLT-fMLLR speech features, the DNN input layer has 440 speech features and n-dimensional additional features **c** as input, where n is the number of additional features used. During training and testing, these additional features are extracted at frame level from the parallel heterogeneous data provided by the vehicles.

3. DATASET

The speech corpus used for our experiments is a 30-hour 2k-vocabulary dataset collected by Ford Motor Company in actual driving, reverberant and noisy environments. The utterances were recorded in vehicles of varying body styles (e.g., small, medium, large car, SUV, pick-up truck) with talkers (drivers) of varying gender, age and dialects, under different ambient noise conditions (blower on/off, road surface rough/smooth, vehicle speed 0-65 MPH, windshield wipers on/off, vehicle windows open/closed, etc.). For our experiments, the data were randomly partitioned into three sets with non-overlapping speakers. The training set contains 17,183 utterances from 90 speakers, the development set contains 2,773 utterances from 14 speakers, and the evaluation set contains 1,763 utterances from 9 speakers. The OOV rate is 5.03%. Aside from the speakers, all other recording conditions are found in all three data sets.

4. EXPERIMENTS

4.1. Speech-based DNN-HMMs

For the baseline speech-only system, we first flat-start trained 26 context-independent monophone acoustic models using MFCC features, then used these models to bootstrap the training of a context-dependent triphone GMM-HMM system. The triphone GMM-HMM system is then retrained by MFCC-LDA-MLLT features. The resulting models contain 3,158 tied triphone states, and 90K Gaussians. This GMM-HMM system is then used to generate fMLLR feature transforms for training and test speakers. The features for both the training and test speakers are then transformed using these fMLLR transforms. The resulting transformed features are input to the neural net. We use the Kaldi toolkit for these experiments [16].

For decoding, a trigram language model with modified Good-Turing smoothing was used. The trigram language model was generated from the 27-hour training data using the sriLM toolkit [17]. The perplexity of the trigram search LM on the Ford development text is 14. Given that we have a small vocabulary task, experiments show that results using a quadgram LM do not differ much with that of a trigram LM. Therefore, trigram LM is used for all following experiments.

The DNN baseline is trained on the fMLLR transformed MFCC-LDA-MLLT features, except that the features are globally normalized to have zero mean and unit variance. The fMLLR transforms are the same as those estimated for the GMM-HMM system during training and testing. The network has 4 hidden layers, where each hidden layer has 1024 units; the DNN has 3158 output units. The input to the network consists of 11 stacked frames (5 frames on each side of the current frame). We perform pre-training using one-step contrastive divergence [18], whereby each layer is learned one at a time, with subsequent layers being stacked on top of the pre-trained lower layers.

For the DNNs in this work, we use two different loss objectives. First we train the 4-layer DNN using back propagation with Cross Entropy (CE) as the objective function. After calculating the gradients for this loss objectives, stochastic gradient descent (SGD) [19] is used to update the network parameters. For SGD, we used minibatches of 256 frames, and an exponentially decaying schedule that starts with an initial learning rate of 0.008 and halves the rate when the improvement in frame accuracy on a cross-validation set between two successive epochs falls below 0.5%. The optimization terminated when the frame accuracy increased by less than 0.1%. Cross-validation is done on a set of 180 utterances that are held out from the training data. The word error rate on the Ford evaluation set is shown in Table 2 as DNN-CE.

Baseline systems	WER
DNN-CE	7.04%
DNN-sequence	6.53%

Table 2. *WER of baseline systems using CE training, followed by sequence training.*

The resulting DNN is then used for sequence training. We use state-level minimum Bayes risk (sMBR) criterion for the sequence-discriminative training. After calculating the gradients for this loss objectives, SGD is used to update the network parameters. The SGD back propagation parameters are the same as with the DNN-CE baseline. The word error rate on the Ford evaluation set is shown in Table 2 as DNN-sequence.

4.2. Heterogeneous DNN-HMMs

In this section we present experimental results of the proposed Heterogeneous DNN-HMM system. We train DNNs augmented with different combinations of the additional features. For each distinct additional feature combination, we train a separate DNN. The network configurations and training/decoding procedures remain the same as in Section 4.1.

In order to compare the effect of different additional features, Table 3 provides the WER of the DNN-CE system using different additional feature combinations. Our experiments show that systems with all different additional feature combinations improved the WERs. Here we only list the result of seven good feature combination candidates in Table 3. Speed gives the lowest WER among the individual additional features. This might be related to the fact that speed is a dominant contributor of the noise. With more additional features included, the WER continues to drop. The complete additional feature set of using speed, ac_fan, wiper_status and vehicle_type gives the lowest WER among all additional feature combinations. This demonstrates that having this information is helpful for noise robustness.

Additional Features (AFs)	WER
+ speed	6.71%
+ ac_fan	6.72%
+ wiper	6.81%
+ vehicle	6.89%
+ speed, ac_fan	6.63%
+ speed, ac_fan, wiper	6.61%
+ speed, ac_fan, wiper, vehicle	6.60%

Table 3. *WER of the DNN-CE system with different additional features.*

Table 4 compares the performance of the baseline systems against our systems with additional features. The additional features used here are $\mathbf{c} = ($speed, ac_fan, wiper, vehicle$)$. We compare the WER with and without the additional features. We can see that, compared to the the DNN-CE baseline, WER is reduced by 6.3%. Compared to the DNN-sequence baseline, WER is reduced by 5.5%. The absolute gain doesn't seem to be huge due to our comparatively small baseline WER, but the 6.3% comparative gain suggests that if this approach is used on a larger-vocabulary task, the absolute gain might be more substantial.

To further demonstrate the effectiveness of the heterogeneous data, in Table 5 we provide the noise-adaptive training (NAT) results using signal-to-noise ratios (SNRs) derived from speech signals. SNRs are computed using the NIST metric. The results demonstrate that additional features we use outperform the SNRs computed from the the speech signals in both CE and sequence training cases. This indicates that the heterogeneous data contain richer information about the environment than the SNRs computed from the speech signals. These additional features are better alternatives to SNR to do noise-adaptive training or environment-aware training. Moreover, the additional features and SNRs are not exclusive. Maybe using them jointly will lead to a better adaptation scenario.

	without AFs	with AFs	WERR
DNN-CE	7.04%	6.60%	6.3%
DNN-sequence	6.53%	6.17%	5.5%

Table 4. *WER comparison of with v.s. without the additional features, and the Word error reduction rate (WERR).*

	with AFs	with SNR
DNN-CE	6.60%	6.91%
DNN-sequence	6.17%	6.51%

Table 5. *WER comparison of using additional features v.s. using SNRs*

5. CONCLUSION

We have shown that DNNs can adapt to environment characteristics if we augment standard acoustic features by appending features from heterogeneous data. It learns useful relationship between these heterogeneous data and noisy speech features, which enable the model to generate more accurate posterior probabilities. This was motivated by the success of noise-aware training where SNRs have been found to be useful for noise robustness because it can serve to characterize the noise level.

Our experiments demonstrate that WER can be reduced by 6.3% with the additional features compared to the baseline DNN-HMM hybrid system. Moreover, this outperforms the improvement brought by noise-aware training using SNRs by a large margin. This indicates that the heterogeneous data contain richer information about the environment than the SNRs. If other heterogeneous data and representations contain similar information about the environment, they can possibly also be used to do environment-adaptation of DNN-HMM system in the same way for better noise robustness.

This framework can also be generalized to incorporate other features, both continuous and discrete, for various robust ASR tasks. For example, visual information, acoustic sensor data and machine status can be explored using this approach. Moreover, different DNN structures can also be investigated, by feeding the additional features at different layer levels into the network.

6. REFERENCES

[1] C. Nikias and J. Mendel, "Signal processing with higher-order spectra," *Signal Processing Magazine*, vol. 10, no. 3, pp. 10–37, 1993.

[2] R. A. Wiggins, "Minimum entropy deconvolution," *Geoexploration*, vol. 16, no. 1, pp. 21–35, 1978.

[3] Z. Koldovskỳ, J. Málek, J. Nouza, and M. Balık, "Chime data separation based on target signal cancellation and noise masking," *Proc. CHiME*, 2011.

[4] K. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. ICASSP*, 2008.

[5] F. Weninger, M. Wollmer, J. Geiger, B. Schuller, J. Gemmeke, A. Hurmalainen, T. Virtanen, and G. Rigoll, "Non-negative matrix factorization for highly noise-robust asr: To enhance or to recognize?," in *Proc. ICASSP*, 2012.

[6] H. K. Maganti and M. Matassoni, "An auditory based modulation spectral feature for reverberant speech recognition.," in *Proc. INTERSPEECH*, 2010.

[7] O. Vinyals and S. V. Ravuri, "Comparing multilayer perceptron to deep belief network tandem features for robust asr," in *Proc. ICASSP*, 2011.

[8] J. Du and Q. Huo, "A feature compensation approach using high-order vector taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2285–2293, 2011.

[9] F. R. Astudillo, *Integration of short-time fourier domain speech enhancement and observation uncertainty techniques for robust automatic speech recognition*, Ph.D. thesis, Technische Universit at Berlin, 2010.

[10] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[11] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 7398–7402.

[12] Arun Narayanan and DeLiang Wang, "Joint noise adaptive training for robust automatic speech recognition," *Proc. ICASSP*, 2014.

[13] Ramesh A Gopinath, "Maximum likelihood modeling with gaussian distributions for classification," in *Proc. ICASSP*. IEEE, 1998, vol. 2, pp. 661–664.

[14] Mark JF Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[15] Spyros Matsoukas, Rich Schwartz, Hubert Jin, and Long Nguyen, "Practical implementations of speaker-adaptive training," in *DARPA Speech Recognition Workshop*. Citeseer, 1997.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[17] Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash, "Srilm at sixteen: Update and outlook," in *Proc. ASRU*, 2011, p. 5.

[18] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, 2009.