NOISE CONTEXT MODEL FOR ROBUST SPEECH RECOGNITION

Suyoun Kim

F-14, 11-751, Speech Recognition and Understanding
Carnegie Mellon University
suyoun@cmu.edu

1. Introduction

In this work, we investigated the effect of incorporation of background noise information into the acoustic model on the robustness of speech recognition system. We evaluated this approach in condition with mismatch between the training and testing data, by combining various addictive noises. The performance can be improved by appending even only 3-dimensional information about the environment into acoustic model.

1.1. Oracle noise context vector

According to SNR and noise type, we manually create 3 dimensional noise context vector as described in Table 1. Then, we append this context vector to each feature frame.

1.2. Bottleneck noise context vector

Noise context vector is learned and extracted from another DNN rather than manually created. We train a DNN which is to classify noise types and SNR instead of triphone states. This DNN has a bottleneck layer which has 50-dimensional bottleneck feature vector at each frame. This vector represents not only discrimination between noise types, but also the interaction between noise and speech signal. Similar to the appending oracle noise context vector, we append the bottleneck feature vector to the original feature vector. Then, we train the final acoustic model with the concatenated feature vectors.

2. Experiments

2.1. Dataset

The experiments were carried out on the DARPA Resource Management (RM) database. The recognition system was trained on 1,600 utterances of clean or degraded speech for training and was tested with 600 utterances.
We generated three additional noisy datasets in our work: (1) digitally-added white noise, (2) digitally-added noise that had been recorded live on urban streets, and (3) digitally-added background music. Each additive noise was included at

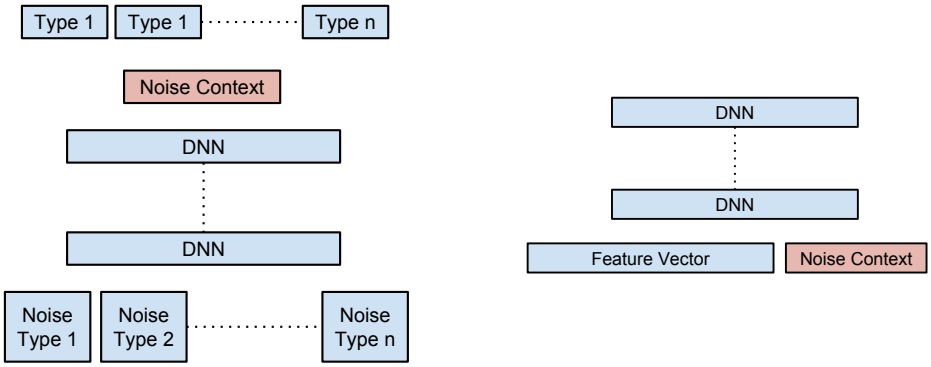

Figure 1: Illustration of the appending oracle noise context vector and bottleneck noise context vector. Left figure describes DNN for extraction of the bottleneck noise context vector, and right figure shows DNN model with noise context vector for ASR.

Noise Type (SNR)	Dim. 1	Dim. 2	Dim. 3
Clean (100)	0	0	0
White (00)	1	0	0
White (05)	0.95	0	0
White (10)	0.9	0	0
White (15)	0.85	0	0
White (20)	0.8	0	0
Street (00)	0	1	0
Street (05)	0	0.95	0
Street (10)	0	0.9	0
Street (15)	0	0.85	0
Street (20)	0	0.8	0
Music (00)	0	0	1
Music (05)	0	0	0.95
Music (10)	0	0	0.9
Music (15)	0	0	0.85
Music (20)	0	0	0.8

Table 1: Manually created 3 dimensional noise context value

Feature (Acoustic Model)	Baseline	With Oracle Noise Context Vector
MFCC (GMM)	26.50	25.29
PLP (GMM)	26.67	26.24
FBANK (GMM)	25.84	25.55
PNCC (GMM)	23.26	23.11
MFCC (DNN)	18.36	17.88
PLP (DNN)	18.51	18.57
FBANK (DNN)	18.33	18.23
PNCC (DNN)	**17.24**	**17.17**

Table 2: WER results of combinations of features (MFCC, PLP, FBANK, and PNCC) and acoustic models (GMM, and DNN) in combined condition with/without Noise Context Vector. The bold fonts represent the lowest WER.

different SNR, 20 dB, 15 dB, 10 dB, 5 dB, and 0 dB for generating the noisy conditions to be a progressively difficult. To evaluate the approach in mismatched condition, the model was trained the combined all noise types with different SNR and then tested with the combined all noise test sets.

2.2. Front-end Features

We investigate the noise robustness performance of the various state-of-the-art front-end features, PNCC, fbank, as well as traditional features, MFCC, PLP using GMM and DNN based acoustic models.

2.3. DNN Architecture

We employ the Kaldi toolkit as ASR back-end system using the RM bigram language model and about 1,500 context-dependent triphone-states in the acoustic model. In a first step, we set up a GMM-HMM baseline system with 13 mean and variance normalized MFCCs with Delta and acceleration, followed by +-3 frame splicing, linear discriminant analysis (LDA), maximum likelihood linear transform (MLLT), and feature-space maximum likelihood linear regression (fMLLR). For the fair comparison, the alignment of the training data to the HMM states is then extracted from the clean training data using MFCC and used later for training on noisy dataset with other features of the DNN-HMM system. The hybrid DNN-HMM Kaldi system is based on "Dan's implementation" using a maxout network with 2-norm nonlinearities/activation functions and 2 hidden layers, each one with an input dimension of 1000 and an output dimension of 200.

3. Conclusions

Through a series of experiments on the the RM database task, we show that appending only 3-dimensional information about the environment to feature vector improves WER performance in environmental distortion and multi-condition training case. Since the bottleneck context feature is higher dimension which can capture the environment information

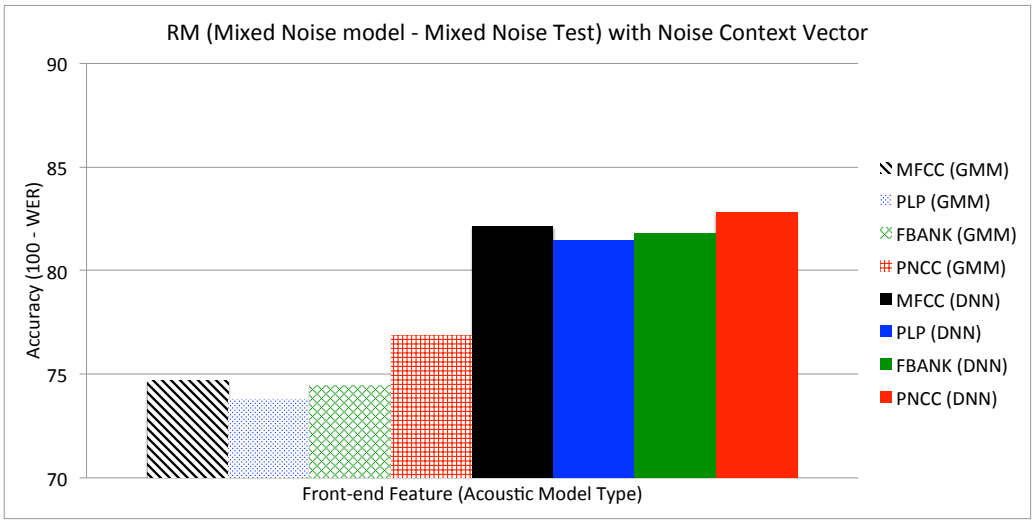

Figure 2: Comparison of recognition accuracy for MFCC, PLP, FBANK, and PNCC based on GMM and DNN using the RM corpus in combined training

sufficiently, we expect that it improves accuracy further.