T31 System Description for SdSV Challenge 2020

T31

1. Introduction

This document provides a brief description of the JXNU system for the Short-duration Speaker Verification (SdSV) Challenge 2020.

2. Feature

For speech parameterization, we extract 30-dimensional MFCCs (including c0) from 25 ms frames every 10 ms using a 36-channel mel-scale filterbank spanning the frequency range 20 Hz–7600 Hz. Before dropping the non-speech frames using an energy based SAD, a short-time cepstral mean subtraction is applied over a 3-second sliding window.

3. Systems

Two sub-systems are applied in our system.

3.1. I-vector/PLDA

The 90-dimensional MFCC feature (delta and acceleration appended) vectors are used to train the UBM with 2048 Gaussians with full covariance matrices. T matrix with rank of 600 is trained on VoxCeleb2 datasets. The i-vector system is trained without augmentation. Length normalization and LDA dimensionality reduction to 200 dimensions followed by another length normalization is applied to i-vectors. All data were centered using the training data mean. For backend scoring, a Gaussian PLDA model with a full-rank Eignevoice subspace is trained. LDA and PLDA are trained on VoxCeleb1 and VoxCeleb2 dataset.

For a single CPU (single threaded), it takes about 200 hours for training the UBM, and 600 hours for training the i-vector exactor. After that, a speech segment takes about one second to extract its i-vector. Training the LDA and PLDA is very quick, and it only takes some minutes. For all trials, after all test speech i-vectors are extracted, it takes only a few minutes to get the final scores.

3.2. X-vector/PLDA

The dataset used in this subsystem is the same as the I-vector/PLDA subsystem. The features are 30-dimensional MFCCs. The x-vector system is built using Kaldi. The following figure shows a block diagram of the x-vector system.

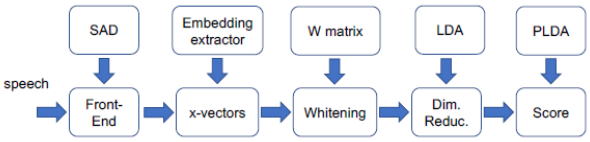

Figure 1: *Block diagram of the x-vector system.*

For x-vector extraction, an TDNN with 7 hidden layers and rectified linear unit (RELU) non-linearities is trained to discriminate among the training set. The first 5 hidden layers operate at frame-level, while the last 2 operate at segment-level. There is 1500-dimensional statistics pooling layer with between the frame-level and segment-level layers that accumulates all frame level outputs from the 5th layer and computes the mean and standard deviation over all frames for an input segment. After training, embeddings are extracted from the 512- dimensional affine component of the 6th layer (i.e., the first segment-level layer).

In order to increase the diversity of the acoustic conditions in the training set, a 5-fold augmentation strategy is used that adds four corrupted copies of the original recordings to the training list. The recordings are corrupted by either digitally adding noise (i.e., babble, general noise, music) or convolving with simulated and measured room impulse responses (RIR).

Prior to dimensionality reduction through LDA (to 200), 512-dimensional x-vectors are centered and unit-length normalized. The centering statistics are computed using VoxCeleb1 and VoxCeleb2. For backend scoring, a Gaussian PLDA model with a full-rank Eignevoice subspace is trained using the x-vectors extracted from all speech segments, as well as one corrupted version randomly selected from {babble, noise, music, reverb}.

It takes about 200 hours for training the x-vector extractor on the GeForce GTX 1050Ti GPU. After that, a speech segment takes about one second to extract its x-vector on a single CPU (single threaded). Training the LDA and PLDA is very quick, and it only takes some minutes. For all trials, it takes only a few minutes to get the final scores after all test speech i-vectors are extracted.

3.3. Score fusion

Finally, the linear logistic regression is used in the score-level system fusion.

4. Performances

The following table shows the results on task1 and task2.

Table 1: *Results.*

Task	EER(%)	MinDCF
Task1	9.70/9.63	0.5352/0.5364
Task2	10.04/10.03	0.4194/0.4199