JHU Sys4 Description

1 Abstract

JHU Sys4 utilized x-vectors trained on 16kHz microphone data. Marks from clustered segments were also refined with VB diarization. In Track 2, SAD marks were determined with a TDNN originally trained on 16kHz speech with augmentations, but with the final layer retrained on the dev data.

2 Data Resources

The SAD algorithm used audio from European Parliament videos. The clean, 16kHz microphone recordings were subsequently augmented with noise and music from the MUSAN corpus, and reverberated with impulse responses from the AIRS corpus.

The x-vector system was trained with data from VoxCeleb, Mixer 4/5, speaker-labeled segments from numerous broadcast corpora (LDC: 97S44, 98S71, 98S73, 2009S02, 2012S06, 2013S02, 2013S04, 2013S07, 2013S08, 2014S07, 2014S09, 2015S01, 2015S06, 2015S11, 2015S13, 2016S01, 2016S03, 2016S07, 2017S02, 2017S15, 2017S25), and the same European Parliament audio used in the SAD training. The x-vector was trained in accordance with the SRE16 Kaldi recipe, only differing in the embedding dimension of 256. PLDA was trained with VoxCeleb only, and all x-vectors both in PLDA training and at test-time were whitened with statistics from the aggregate of VoxCeleb, Mixer 4/5, and the DIHARD data.

The UBM and T matrix used in the resegmentation algorithm was trained on VoxCeleb only.

3 Algorithm Details

3.1 SAD

The SAD algorithm was a 5-layer TDNN with a final sigmoid layer and used log-compressed magnitudes from 35 mel filters after a 3-second sliding mean subtraction as input. The first layer aggregated 5 frames (50ms) of context, and each subsequent layer doubled the width of a 3 tap filter, resulting, in the end, in +/- 640 ms of context at the final sigmoid layer. ReLU non-linearities were used between all layers (except the final sigmoid non-linearity). While the system was initially trained on the European Parliament audio (with a cross-entropy metric), the final layer was retrained using the DIHARD dev data. Final output probabilities were smoothed with a 500ms median filter prior to thresholding at 0.50.

3.2 Segment Clustering

Speech was segmented into approximately 1.5 second windows with 0.75 second hops, 24 MFCCs were extracted every 10ms, and a 256-dimensional x-vector was extracted for each segment. These segment x-vectors were scored with PLDA (trained with segments labeled only for speaker) and clustered with AHC (average score combination at merges). The stopping threshold for merging was learned on the dev data and tuned to optimal DER.

3.3 Resegmentation

Resegmentation was performed with VB diarization [1]. The input features were 24 MFCCs computed every 10ms. The algorithm was initialized with the marks from clustering, then allowed to run only one pass with a downsample parameter of 3.

4 System Requirements

- The x-vector training was distributed across numerous GPUs, in accordance with the linked Kaldi recipe.
- The SAD system was built in PyTorch and trained on a single GeForce GTX 1080 GPU card with 12GB of available memory.
- All elements were run on CPUs only at test time (again determined by scheduler with no concern for CPU specifics).
- **To process 10-minute recording**: 79 seconds (12 seconds for SAD marks, 60 seconds for clustering, 7 seconds for resegmentation)

[1] http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors