# Team08's System Description for SdSV Challenge 2020

# (Text-independent: Task2)

*TEAM08*

## Abstract

This report is about the TEAM08's submission to the Short-duration Speaker Verification (SdSV) Challenge 2020 text-independent (TD) verification TASK 2. Our approach for the challenge consists of the fusion Kaldi's two x-vectors and two i-vectors subsystems.

## 1. Train and Development Set

Task 2 of the SdSV Challenge 2020 [1, 2, 3] is defined as speaker verification in text-independent mode: given a test segment of speech and the target speaker's enrollment data, automatically determine whether the test segment was spoken by the target speaker.

For development (Dev) data set task 2 provided by the SdSV challenge [1, 2, 3], we choose odd number speakers (total 294 speakers) from the train dataset as dev enroll and dev test. Random selected three utterances from each speaker are used to train a targeted speaker model. Total 5586 speaker models are built. The remaining dev data set (even number speaker utterances total 294 speakers) are used the background dataset, such as for the x-vector or i-vector LDA and PLDA. Table 1 shows the DEV dataset for task 2.

Table 1: SdSV TASK2 DEV dataset design

| | No spks | No Spks model | Total Trials | Target Trials | Imposter trials |
|---|---|---|---|---|---|
| Dev Set SdSV train | 294 (odd num.) | 5586 | 2463720 | 111720 | 2352000 |

We adopted the Kaldi's SITW [4, 5] and Voxceleb [6] x-vector and i-vectors as one of subsystems. For the Kaldi system, we use the following training data for the model training: VoxCeleb 1 & 2, MUSAN [7] (Noise and music dataset) and RIRS noise [8] for x-vector and i-vector model training. In addition, The Voxceleb 1 & 2 datasets are also used to train LDA model and out-of-domain PLDA backend. And the 294 even number speakers utterances extracted from SdSV train set are used to design in-domain PLDA model backend. The details usages of datasets are summarized in Table 2.

Table 2: Data Usages for the Model Training

| Usages | Dataset |
|---|---|
| x/i-vector models | Voxceleb 1&2 MUSAN (noise and music), RIRS. |
| LDA, out-of-domain PLDA | Voxceleb 1 & 2 |
| in-domain PLDA, LDA, PLDA, | SdSV training 294 even number speakers utterances extracted from SdSV train set |

## 2. Kaldi Systems

We used the Kaldi SITW/Voxceleb 16khz x-vector recipe [4]. The Voxceleb1 and Voxceleb2 dataset [6] were used for the TDNN background model. The even number utterances for dev training are used for LDA and PLDA. We extracted the utterances related x-vectors or i-vectors from all the subset of the dev-train as the LDA. The same speaker related utterance is used to build a PLDA speaker cluster.

## 3. Experimental Results

In this Section, simple experimental results are reported. Since the evaluation dataset x-vector system is worse than i-vector system. There have something wrongs for our approach, although the dev dataset results are quite consistent with the expectation. The results are summarized in Table 3. No further experiments are conducted.

The Bosaris toolkit [9] is used to do the linear fusion. The SdSV Dev subset trials are used for the parameters tuning.

Table 3. The Kaldi system results

| | DEV | | **EVAL SETS** | |
|---|---|---|---|---|
| | EER*(%)* | *min*DCF | **EER(%)** | ***min*DCF** |
| SITW-ivector | 1.138 | 1.06 | 9.59 | 0.388 |
| SITW-xvector | 0.863 | 0.0835 | 11.15 | 0.465 |
| Voxceleb-ivector | 1.169 | 0.107 | 9.7523 | 0.392 |
| Voxceleb-xvector | 0.993 | 0.0883 | 10.68 | 0.452 |
| **Fusion all** | **0.653** | **0.0560** | **8.48** | **0.3398** |

## 4. Real-Time Factor and Memory Usage

The CPU real-time factor (RT) and memory usages are listed in Table 4 based on Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz.

Table 4: Real-Time Factor and Memory Usage

| TASKS | RT or Memory |
|---|---|
| SITW x-vector | 0.31 |
| SITW i-vector | 0.32 |
| Voxceleb x-vector | 0.31 |
| Voxceleb i-vector | 0.32 |
| Memroy usage | 300 (peak) |

# 5. References


[1] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, Luka˘s Burget, "Short-duration Speaker Verification (SdSV) Challenge 2020: the Challenge Evaluation Plan" https://sdsvc.github.io/assets/SdSV_Challenge_Evaluation_Plan.pdf

[2] H. Zeinali, H. Sameti, and T. Stafylakis, "DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English." in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 386–392.

[3] H. Zeinali, L. Burget, and J. Cernocky, "A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: the DeepMine database," in *Proc. ASRU 2019 The 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.

[4] Kaldi ASR tools kits, Available: https://github.com/kaldi-asr/kaldi.git.

[5] SITW Dataset, Available: http://www.speech.sri.com/projects/sitw/.

[6] Voxceleb data, http://www.robots.ox.ac.uk/ vgg/data/voxceleb/

[7] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," CoRR, vol. abs/1510.08484, 2015.

[8] RIRS noise, http://www.openslr.org/resources/28/rirs_noises.zip.

[9] N. Brummer and E. de Villiers, "The Bosaris toolkit," Available: https://sites.google.com/site/bosaristoolkit/.