I2R's System Description to Short-duration Speaker Verification (SdSV) Challenge 2020
Text-dependent: Task1

Hanwu Sun, Kah Kuan Teh and Huy Dat Tran

Institute for Infocomm Research, A*STAR, Singapore

{hwsun, teh_kah_kuan, hdtran}@i2r.a-star.edu.sg

Abstract

This report is about the I2R's submission to the Short-duration Speaker Verification (SdSV) Challenge 2020 text-dependent (TD) verification. Our approach for the challenge consists of the fusion Kaldi's two x-vectors and two i-vectors, GMM-SVM MFCC subsystem and GMM-SVM BNF subsystem. Main efforts have been focused on the pass-phase verification, PLDA/NAP backend design and system fusion studies in order to improve the system performance for the test-dependent speaker verification. DNN posterior and DTW methods are applied to conduct the pass-phrase verification. we have contributed pass-phase based PLDA or NAP system to enhance the system performance.

Index Terms: speaker recognition, text-dependent, text-independent, DNN

1. Introduction

Task 1 of the SdSV Challenge 2020 [1, 2, 3, 4] is defined as speaker verification in text-dependent mode [1, 4, 5]: given a test segment of speech and the target speaker's enrollment data, automatically determine whether a specific phrase and the test segment was spoken by the target speaker.

For development (Dev) data set provided by the SdSV challenge [1, 2, 3, 4], we choose odd number speaker from the train dataset as dev enroll and dev test. The remaining dev data set (even number speakers' utterances) are used the background dataset, such as for the x-vector or i-vector LDA, PLDA and GMM-SVM Nuisance Attribute Projection (NAP) for the channel compensation. In the addition, we also use the even number speaker utterances in the training dataset to build the pass-phrase speaker independent models for the pass-phrase verification.

Table 1: SdSV DEV dataset design

	No spks	No utts	Total Trials	TC/IC Trials	TW/IW Trials
Dev Set SdSV train	482 (odd num.)	51849	6935048	38500/ 3864000	1489908/ 1542640

For the text-dependent speaker verification, we need to consider four test trials scenarios. Target speaker-Correct pass-phrase trials, Imposter speaker-Correct pass-phrase trials, Target speaker-Wrong pass-phrase trials and Imposter speaker-Wrong pass-phrase trials. We simplify them as TC, IC, TW and IW, respectively. So, there will have three type imposter trials. As we know Target speaker-Wrong pass-phrase trials (TW) is correct trials in the text-independent speaker recognition. Based on the above definition, the following table shows our designed DEV dataset used for our SdSV's task 1 evaluation.

We adopted the Kaldi's SITW [6, 7] and Voxceleb [8] 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 [9] (Noise and music dataset) and RIRS noise [10] 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 481 even number speakers utterances extracted from SdSV train set are used to design in-domain PLDA model backend. Such data set are also used as the NAP for GMM-SVM system [11]. Subset of SdSV training utterances is also used for the score normalization. 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, GMM, LDA, PLDA, NAP	SdSV training 481 even number speakers utterances extracted from SdSV train set
Score normalization	SdSV training subset utterances

2. Bottleneck Feature and DNN Posterior

In this challenge, we follow the implementation described in [6, 12] to generate BNF [13]. The Kaldi's script [6] and Librispeech [14] clean datasets are used to train the model.

Once the DNN is trained, the linear outputs of bottleneck layer are extracted to 64 dimensions BNF. Meanwhile, we also use the model to extract all the SdSV's task 1 utterance DNN posteriors, which will be used for the pass-phrase verification next.

3. Pass-Phrase Verification

For the text dependent speaker recognition, the pass-phrase verification is one of the key complements. Normally, the DNN pass-phrase verification [15, 16] is to compute distances between target and test speaker utterances by using their DNN posterior after applying DTW alignment. The DTW algorithm takes two sequences as input and matches their content by finding the path with the smallest alignment between them.

One of the problems for such directly approach is the computation cost. Each test trial will do the computation between model utterances and testing utterance. Another

problem is how to choose the decision threshold. Such threshold may vary with different utterances and noisy condition. In the SdSV's challenge, there are ten fixed pass-phrases [1, 2].

We build speaker-independent pass-phrase DNN posteriors from Dev train set (using 481 even dev-train speaker related pass-phrase utterances). We cannot use all dev training utterances to do the pass-phrase verification due to huge testing and comparing costs. Here we build a speaker dependent pass-phrase verification method. we only select partial pass-phrase DNN posterior to build common pass-phrase DNN posterior.

The detailed experimental results will be presented in Section 5.

4. Kaldi and GMM-SVM Systems

Authors must proof read their PDF file prior to submission to ensure it is correct. Authors should not rely on proofreading the Word file. Please proofread the PDF file before it is submitted.

4.1 Kaldi's Based Systems

we used the Kaldi SITW/Voxceleb 16khz x-vector recipe [6]. The Voxceleb1 and Voxceleb2 dataset [8] were used for the DNN background model. Then, we extracted the utterances related x-vectors or i-vectors from all the subset of the dev-train as the LDA. The PLDA is built in the two methods:

A. First one is to use all the dev-train even pass-phase x-vector as the PLDA. The same speaker and same pass-phrase are grouped to one cluster to compute the PLDA model. Symmetric score normalization (S-norm) is selected as the cohort that contains 11000 utterances x-vectors or i-vectors random selected from dev-train subset, which contains all 10 pass-phrase utterance x-vectors or i-vectors.

B. Another one is pass-phrase dependent PLDA. Only same pass-phrase speaker model related trials are modeling and testing. The LDA is the same as the above. S-norm uses the cohort that only contains the same pass-phrase utterances random selected from dev-train subset. 2000 utterances for each pass-phrase are selected for each phase-phrase. We select top 10% scores for both these two methods

4.2 GMM-SVM System

Based on the MFCC (50 dims), BNF (64 dims) and MFCC combined BNF tandem (114 dims) features, we use GMM-SVM to build these three subsystems.

The GMM-SVM system uses these three features separately to extract their related supervectors to construct kernels of support vector machines (SVMs). The 256 Gaussian mixture component models are trained. The means of the GMM mixture components is subtracted to construct the supervector by extracted its mean and normalized by the unit norm.

As a result, the 64 dimensions bottleneck feature was expanded into 16384 dimension supervector. 50 dimension MFCC into 12800 dimension supervectors and the tandem feature into 29184 dimension supervectors. For each supervectors, we build two schemes for channel or pass-phrase compensation. Similar to Kalid's PLDA approach, we have used two NAP approaches for the pass-phrase compensation

A. One is to use all the speaker related pass-phrase cluster to build a NAP, namely: global NAP, which includes all the ten dev-train pass-phrases. Similarly, the same speaker and the same pass-phrase utterances are extracted to form a NAP cluster.

B. Another one is to use individual pass-phrase speaker supervectors to build pass-phrase dependent NAP.

Score normalization (Z-norm) [11, 12] is conducted for these GMM–SVM systems. We have random selected 1000 utterances from train-dev dataset for global GMM-SVM system to do the z-norm. For the pass-phrase dependent GMM-SVM approach, only pass-phrase related 800 utterances from dev-train set are selected to do the z-norm,

5. Experimental Results

In this Section, the detailed experimental results are provided. We firstly demonstrated the effect of pass-phrase verification on the EER and *min*DCF differences before and after applying the pass-phrase verification. In Section 5.2, we have given the detailed results of four subsystems: namely, Kaldi's based four subsystems, GMM-SVM MFCC, GMM-SVM BNF system and GMM-SVM tandem subsystem. The comparison results of globale PLDA or NAP approach and Pass-phrase PLDA or NAP are also provided. Finally, the fusion results are showed in Section 5.3. the designed DEV dataset and Evaluation EER and MinDCF are used to do the comparisons. Since the EER and *min*DCF results of 30% test progress set and 70% test evaluation set are similar, especially for the *min*DCF values. we only report the test 70% evaluation set for the score analysis.

We use the Bosaris toolkit [17] to do the fusion in our submission. The proposed SdSV Dev subset trials are used for the parameters tuning. In this challenge, we submitted 2 scores: one is the primary fusion score and another one is the best single system to demonstrate the fusion effects.

5.1. Pass-Phrase Verification Results

One of major challenge is to detect Target Speaker-Wrong-pass-phrase (TW) related trials for the pass-phrase speaker recognition. Meanhile, we can use the TC-TW performance to evaluation the performance of the pass-phrase verification. In such way, we can compute the performance of the pass-phrase in the following two ways:

1. Pass-phrase verification accuracy: we use dev enroll and testing data to check the overall pass-phrase identification accuracy. Since each speaker model uses three same pass-phrase utterances to do the enroll. We can have two pass-phrase verification identification rates. One is the model enroll pass-phrase identification rate and another is thed single testing utterance identification rate.
2. TC-TW speaker verification performance: We can also compute TC-TW related trials EER and *min*DCF to demonstrate the importance of the Pass-phrase verification.

We achieved 100% identification rate for the dev set enroll model and 99.876% for all the dev test utterances. To show the speaker verification TC-TW trial subtask performance and our dev and evaluation tasks, we used our single best GMM-SVM Tandem system to show the effect of the pass-phrase verification. Table 3 and Table 4 show the performances of TC-TW subtasks and the DEV/Eval whole trials tasks.

Table 3. **TC-TW** subtask EER and *min*DCF

	Dev dev		**Eval (70%)**	
	EER*(%)*	*min*DCF	**EER(%)**	***min*DCF**
No Verif.	46.93	1	***NA***	***NA***
with Verif.	0.004	0.0004	0.01	0.0001

Table 4 Dev and Eval Data Tandem SMM-SVM Overall Performance.

	Dev		**Eval (70%)**	
	EER*(%)*	*min*DCF	**EER(%)**	***min*DCF**
No Verif.	2.811	0.4204	3.361	0.167
With verif key.	0.967	0.0636	2.346	0.073
True key	0.967	0.0635	***NA***	***NA***

5.2. Results of Four Approached Subsystems

In the section, we presented our four different approaches: Kaldi's SITW/Voxceleb x-vector and i-vector, GMM-SVM MFCC, GMM-SVM BNF and GMM-SVM Tandem systems. The comparison focus on the Global PLDA or NAP and Pass-phrase dependent PLDA or NAP.

The subsystems of each approach is summarized in Table 5, 6, 7 and 8, respectively for Kaldi system, three GMM-SVM systems. All the results are output after applying the pass-phased verification.

Table 5. The Kaldi System DEV Results

SdSV's DEV SET	Global PLDA		**Pass-phrase PLDA**	
	EER*(%)*	*min*DCF	**EER(%)**	***min*DCF**
SITW-ivector	2.118	0.141	1.737	0.136
SITW-xvector	1.506	0.113	1.342	0.105
Voxceleb-ivector	2.113	0.140	1.760	0.132
Voxceleb-xvector	1.506	0.113	1.327	0.104
Fusion all	**1.246**	**0.0843**	**1.004**	**0.0742**

Table 6. GMM-SVM MFCC Subsystems

GMM-SVM MFCC	DEV dataset		Eval dataset (70%)	
	EER*(%)*	*min*DCF	EER*(%)*	*min*DCF
Global NAP	2.503	0.171	3.855	0.142
Pass-phrase NAP	1.914	0.125	3.318	0.115
Fusion	**1.496**	**0.100**	**2.810**	**0.095**

Table 7. GMM-SVM BNF system before

GMM-SVM BNF	DEV dataset		Eval dataset (70%)	
	EER*(%)*	*min*DCF	EER*(%)*	*min*DCF
Global NAP	1.980	0.151	3.324	0.123
Pass-phrase NAP	1.723	0.125	2.994	0.107
Fusion	**1.347**	**0.104**	**2.656**	**0.094**

Table 8. GMM-SVM Tandem system

GMM-SVM Tandem	DEV dataset		Eval dataset (70%)	
	EER*(%)*	*min*DCF	EER*(%)*	*min*DCF
Global NAP	1.019	0.070	2.101	0.067
Pass-phrase NAP	0.967	0.064	2.346	0.073
Fusion	**0.838**	**0.054**	**1.931**	**0.059**

5.3. I2R Final Dev Sets and Submission Eval Sets Results

The I2R final submission consists of the four subsystem linear fusion with DNN Posterior pass-phrase verification by using Bosaris toolkit [17].

In order to demonstrate the complementary effects for our four subsystems, Table 9 shows the Eval dataset and Dev subset (EER) performance progresses starting from Kladi's fusion with fusion of other subsystems to demonstrate the complementary effect of these four subsystems.

Table 9. Fusion results of Different Subsystems Combination.

	Dev dev		Eval (70%)	
	EER(%)	minDCF	EER(%)	minDCF
Kaldi	1.004	0.075	2.426	0.083
Kaldi+GMM MFCC	0.779	0.055	2.094	0.064
Kaldi+GMMBNF	0.695	0.0444*	1.976	0.0554
Kaldi+GMM-SVM/BNF	0.620	0.0406	1.802	0.0499
Final fusion4	**0.573**	**0.0379**	**1.690**	**0.0470**

6. References

[1] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, Lukaˇs 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] H. Zeinali, H. Sameti, L. Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification", IEEE/ACM Transactions on Audio, Speech, and Language Processing 25 (7) (2017) 1421–1435.

[5] A. Larcher, K. A. Lee, B. Ma, H. Li, Text-dependent speaker verification: Classifiers, databases and rsr2015, Speech Communication 60 (2014) 56–77.

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

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

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

[9] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *CoR*R, vol. abs/1510.08484, 2015.

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

[11] H. Sun, K. Lee and B. Ma, "A new study of GMM-SVM system for text-dependent speaker recognition", *ICASSP* April, 2015, pp. 4195-4199, Brisbane.

[12] H. Sun, K. Lee etc, "I2R-NUS Submission to Oriental Language Recognition AP16-OL7 Challenge", APSIPA, Kuala Lumpur, Dec. 2017.

[13] Sibel Yaman1, Jason Pelecanos1, Ruhi Sarikaya, "Bottleneck Features for Speaker Recognition", Odyssey 2012 speaker recognition workshop, pp. 105-108, 25-28 June 2012, Singapore.

[14] LibriSpeech dataset, http://www.openslr.org/12/

[15] Subhadeep Dey, Srikanth Madikeri, Marc Ferras and Petr Motlicek, "DEEP NEURAL NETWORK BASED

POSTERIORS FOR TEXT-DEPENDENT SPEAKER VERIFICATION",

[16] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, M Barrios, "The VOiCES from a Distance Challenge 2019 Evaluation Plan," arXiv:1902.10828 [eess.AS], March 2019.

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