STC SPEAKER RECOGNITION SYSTEM FOR THE NIST CTS 2021 CHALLENGE

Anastasia Avdeeva[1], *Aleksei Gusev*[1], *Yuri Khokhlov*[1], *Maxim Korenevsky*[1], *Mariya Korenevskaya*[1], *Igor Korsunov*[1], *Alexander Kozlov*[1], *Galina Lavrentyeva*[1], *Ivan Medennikov*[1], *Sergey Novoselov*[1], *Timur Pekhovsky*[1], *Tatiana Prisyach*[1], *Andrey Shulipa*[2], *Tatiana Timofeeva*[1], *Alisa Vinogradova*[1], *Vladimir Volokhov*[1]

[1] STC Ltd., St. Petersburg, Russia
[2] ITMO University, St. Petersburg, Russia

{avdeeva-a, gusev-a, khokhlov, korenevsky, korenevskaya, korsunov, kozlov-a, lavrentyeva, medennikov, novoselov, tim, prisyach, shulipa, timofeeva, gazizullina, volokhov}@speechpro.com

ABSTRACT

This paper presents a description of STC Ltd. systems submitted to the NIST 2021 Speaker Recognition Evaluation (SRE) conversational telephone speech (CTS) challenge. These systems include different subsystems based on using different neural networks as speaker embedding extractors. During the NIST CTS Challenge we focused on the training of the state-of-the-art deep speaker embeddings extractors like ResNets and ECAPA networks. We used additive angular margin based loss functions. Additionally, inspired by the recent success of the wav2vec 2.0 features in automatic speech recognition we explored the effectiveness of this approach for the speaker verification. According to our observation the fine-tuning of the pretrained large wav2vec 2.0 model provides our best performing system for the evaluation progress set. The final results for submitted systems were obtained by different configurations of subsystems fusion on the score level followed by score calibration.

Index Terms— Speaker recognition, CTS challenge, Resnet, ECAPA, DTDNN, wav2vec 2.0

1. INTRODUCTION

Today's state-of-the-art [1, 2, 3, 4, 5] speaker recognition systems are based on very deep convolutional neural networks (ResNets, ECAPAs, Extended TDNNs) which take as input Log Mel-filter bank features and trained on large datasets using additive angular margin loss functions and different optimization strategies. The simple cosine or PLDA scoring are usually used as an extractors back-end. Adaptive s-norm is applied to improve systems performance. In our study we decide to follow this principles while developing our systems for the NIST SRE CTS challenge.

The special attention was paid for developing alternative Neural Back-End (NBE) for the deep speaker embeddings extractors.

Inspired by the success of wav2vec 2.0 in speech recognition tasks [6, 7] in our work we performed new study of wav2vec 2.0 model fine-tuning for speaker recognition tasks. It should be noted that wav2vec 2.0 models are powerful transformer based models which use raw speech signals as its input and incorporate multi-head attention mechanism on the deep layers to process information.

This paper presents the detailed description of the systems submitted by STC LTD to NIST CTS 2021 Challenge and its performance estimations on different benchmarks and progress set of the challenge.

2. TRAIN DATASETS

we used a wide variety of different datasets containing telephone and microphone data from private datasets and from those available online:

- Switchboard2 Phases 1, 2 and 3;
- Switchboard Cellular;
- Mixer 6 Speech;
- NIST SREs 2004 - 2010;
- NIST SRE 2018 (eval set);
- concatenated VoxCeleb 1 and 2;
- RusTelecom v2;
- RusIVR corpus.

RusTelecom v2 is an extended versions of private Russian corpus of telephone speech, collected by call-centers in Russia. RusIVR is a private Russian corpus with telephone and media data, collected in various scenarios and recorded by different types of devices (telephone, headset, far-field microphone, etc). All files are sampled at 8 kHz.

In order to increase the amount and diversity of the training data, we used Kaldi augmentation recipe (reverberation, babble, music and noise) with the freely available MUSAN and simulated Room Impulse Response (RIR) datasets.

In total, this training dataset contains 1,679,541 recordings from 33,466 speakers.

3. SYSTEMS

This section contains the description of all single systems used for final submissions.

3.1. Front-End processing

In our study we consider popular Log Mel-filter bank energies for ResNet and ECAPA like architectures and raw speech signals for the wav2vec 2.0 based systems. Brief description of these front-ends goes as follows:

8kHz MFB. We use Log Mel-filter bank (MFB) energies extracted from the raw 8kHz signals using the following settings:

- frame-length - 25 ms
- frame-shift - 10 ms
- low frequency - 20 Hz
- high frequency - 3700 Hz
- number of mel bins - 64

After the features were extracted Mean Normalization (MN) over a 3-second sliding window was applied. The U-net-based VAD was used after the MN-normalization procedure to filter out non-speech segments.

Raw audio signal processing. For our wav2vec based extractors we used raw 16 kHz audio. 8kHz utterances were upsampled to 16kHz by sox utility. Additionally, on-line augmentation scheme was used during the training process for raw audio samples using the following probabilities of noise to be used:

- MUSAN additive noise with $p = 0.25$;
- RIR convolution with $p = 0.25$;
- Frequency masking with $p = 0.25$;
- Time masking with $p = 0.25$;
- Clipping Distortion with $p = 0.25$.

Here p is a probability of applying augmentation type for the sample in the training batch. All considered augmentations were applied in sequence.

3.2. Speaker embedding extractors

During all stages of training and tuning processes AAM-Softmax loss was used with parameters m and s set to 0.35 and 32 respectively.

For training we used One Cycle learning rate scheduler with SGD optimizer. In some cases last tuning steps were performed using Adam and small constant learning rate value.

ECAPA-TDNN This model is based on ECAPA-TDNN [8] architecture with the following parameters: the number of SE-Res2Net Blocks is set to 4 with dilation values 2,3,4,5 to blocks; the number of filters in the convolutional frame layers C is set to 1024 equal to the number of filters in the bottleneck of the SE-Res2Net Block; ASP is used; embedding layer size is set to 512. Stem block changed to stack of 4 Conv2D, BatchNorm2D, ReLU layers with 3 kernel size and 32 filters and last Conv1D with 1024 filters. Model was trained on the 8kHz MFB data in several stages with increasing crop size, loss margin and decreasing learning rate. At the final stage we used long segments (10, 12 seconds) for fine-tuning procedure.

ResNet101. This model is based on ResNet101 architecture with the following modifications:

- Maxout activation function on the embedding layer;
- stride = 1 in the first BotleneckBlock;
- simple Conv2D stem block.

Model training and tuning procedures were the same as described above for ECAPA-TDNN.

ResNet146. This model uses standard ResNet146 architecture. Similar to ResNet101 the model was firstly trained using MFB features for short speech segments (5 seconds) and then tuned iteratively for several epochs using longer speech segments (10, 12 seconds).

DTDNN. This model is our implementation of Densely Connected Time Delay Neural Network [9]. To train the extractor we used recommended settings from the original paper. The model was pretrained on 5 seconds speech chunk duration and then fine-tuned using 12 seconds speech segments.

Wav2vec-TDNN. The main scheme of wav2vec 2.0 based speaker embeddings extractor is represented on Figure 1. As an effective wav2vec 2.0 back-end we applied two TDNN layers (the 1st with ReLU activation), statistic pooling layer to pool time series to single vector, maxout linear layer [10, 3] to get speaker embedding. We used AAM-Softmax based linear classification layer to fine-tune the extractor. In principle, one can pass output of the wav2vec directly to the statistics pooling layer. Our intuition here is as follows. We realised that unsupervised wav2vec model pretraining leads to good speech information generalization on the top layers of the autoregressive model. The role of TDNN layers is to prefilter speaker specific information and to ”prepare” wav2vec output for statistical pooling. According to our observations this approach

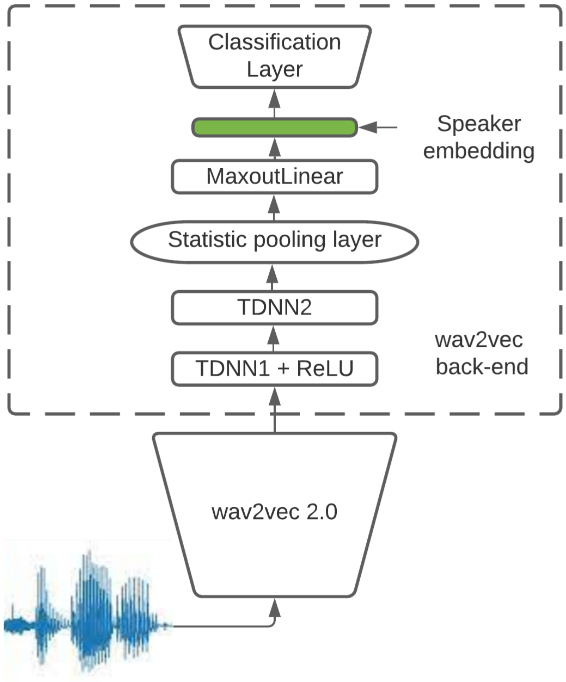

Fig. 1. Wav2vec 2.0 based speaker embeddings extractor

let us achieve better results than direct statistical pooling of the wav2vec outputs. The TDNN blocks utilise context 1 of the input features and have 2048 dimension output. The obtained final speaker embedding size was 512. Additional note is that wav2vec part of the extractor could be freezed while tuning for downstream speaker recognition task. We observed that in this scenario the results can also be very good, but fine-tuning the whole extractors provide additional performance gains for speaker recognition systems.

Our wav2vec-TDNN models are based on wav2vec 2.0 large architecture. We used large multi-lingual wav2vec 2.0 model $XLSR\_53$ provided by facebook [11] on *fairseq cite* as a starting point for the model fine-tuning on 2 dataset.

There are two types of wav2vec-TDNN models in this study. The first one wav2vec-TDNN(clean) was trained on clean version of the dataset 2 . The second one wav2vec-TDNN(aug) was tuned using online augmentation scheme described in Section 3.

3.3. Back-Ends

3.3.1. *Cosine similarity*

We used Cosine similarity to distinguish speaker embeddings:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathbf{x_1}^T \mathbf{x_2}}{\mathbf{x_1} \mathbf{x_2}}, \tag{1}$$

where $(\mathbf{x_1}, \mathbf{x_2})$ are speaker embedding vectors.

3.3.2. *NeuralNet Back-End*

We investigated neural network based back-end (NBE) as an alternative to the conventional cosine similarity scoring. NBE is a simple feedforward neural network that takes two embeddings (test and enroll ones) as an input and outputs a probability that these embeddings correspond to the same speaker. To train NBE, we constructed positive (matching speakers) and negative (non-matching speakers) training examples. A positive example is just a pair of embeddings corresponding to different utterances of the same speaker. On the other hand, a negative example is a pair of embeddings corresponding to different speakers while having a high cosine similarity score. Due to the nature of the DCF metric, we balanced training examples in a ratio of 8 negative per 1 positive. Performance of NBE in terms of EER and DCF was competitive to the cosine similarity. Moreover, a combination of NBE and cosine similarity scores provided significant improvement over a single back-end.

We also utilized NBE to fuse several types of speaker embeddings. For that, we designed a single neural network with four similar branches corresponding to four types of speaker embeddings described in subsection 3.2, namely ECAPA-TDNN, ResNet101, ResNet146, and DTDNN. Each branch takes a pair of embeddings as input, and then outputs of these branches are fed to a combining layer followed by an output layer. Such NBE achieved much lower EER and DCF, comparing to a simple fusion of cosine similarity scores.

3.3.3. *Class posteriors logit embeddings*

During our investigation we observed that the extractors classification layer outputs (namely class posteriors logit embeddings, or cl-embeddings) could be more informative than conventional pre-last layer embeddings. We realised that top linear layer obtained in closed classification task during discriminative training could contain useful information for open task speaker recognition. We explored some naive ideas of using this information by doing speaker verification on the classification layer output with cosine similarity metric scoring.

3.3.4. *Score normalization and calibration*

For all systems adaptive scoring normalization technique (adaptive s-norm) from [12] was used. Here the normalized score for a pair $(\mathbf{x_1}, \mathbf{x_2})$ can be estimated as follows:

$$\hat{\mathcal{S}}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) - \mu_1}{\sigma_1} + \frac{\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) - \mu_2}{\sigma_2}, \tag{2}$$

where the mean μ_1 and standard deviation σ_1 are calculated by matching $\mathbf{x_1}$ against impostor cohort and similarly

Table 1. Results of cl-embeddings, scores normalization and fusion of the different systems prepared for the NIST SRE 21 CTS challenge. The results obtained for NIST SRE 18 dev set, NIST SRE 19 and NIST SRE 16 eval sets.

System	**Dataset**	**cl-emb**	**s-norm**	**EER**	**DCF(0.05)**	**DCF(0.01)**
ResNet101	nist2018_dev	-	-	4.00	0.157	0.234
	nist2016_eval	-	-	6.84	0.408	0.597
	nist2019_eval	-	-	2.78	0.162	0.268
ECAPA-TDNN	nist2018_dev	-	-	4.75	0.186	0.27
	nist2016_eval	-	-	9.61	0.564	0.832
	nist2019_eval	-	-	3.33	0.196	0.329
ResNet101	nist2018_dev	✓	-	3.66	0.14	0.206
	nist2016_eval	✓	-	5.96	0.347	0.52
	nist2019_eval	✓	-	2.50	0.145	0.248
ECAPA-TDNN	nist2018_dev	✓	-	4.63	0.173	0.256
	nist2016_eval	✓	-	8.72	0.525	0.8
	nist2019_eval	✓	-	3.13	0.18	0.305
ResNet101	nist2018_dev	-	✓	3.44	0.13	0.205
	nist2016_eval	-	✓	5.64	0.274	0.409
	nist2019_eval	-	✓	2.52	0.145	0.24
ECAPA-TDNN	nist2018_dev	-	✓	4.28	0.167	0.246
	nist2016_eval	-	✓	8.88	0.385	0.533
	nist2019_eval	-	✓	3.07	0.177	0.291
ResNet101	nist2018_dev	✓	✓	3.42	0.128	0.194
	nist2016_eval	✓	✓	5.01	0.237	0.357
	nist2019_eval	✓	✓	2.39	0.134	0.228
ECAPA-TDNN	nist2018_dev	✓	✓	4.20	0.154	0.228
	nist2016_eval	✓	✓	8.59	0.337	0.465
	nist2019_eval	✓	✓	2.97	0.165	0.269
ResNet101 + ECAPA-TDNN	nist2018_dev	✓	✓	3.20	0.115	0.168
	nist2016_eval	✓	✓	4.87	0.221	0.328
	nist2019_eval	✓	✓	2.12	0.122	0.214
DTDNN	nist2018_dev	✓	✓	4.37	0.1626	0.2243
	nist2016_eval	✓	✓	7.65	0.4158	0.595
	nist2019_eval	✓	✓	3.33	0.196	0.322
ResNet146	nist2018_dev	✓	✓	3.60	0.141	0.207
	nist2016_eval	✓	✓	5.41	0.244	0.36
	nist2019_eval	✓	✓	2.78	0.154	0.253
wav2vec-TDNN(clean)	nist2018_dev	✓	✓	3.22	0.097	0.148
	nist2016_eval	✓	✓	**3.87**	**0.193**	**0.287**
	nist2019_eval	✓	✓	**1.76**	**0.102**	**0.187**
wav2vec-TDNN(aug)	nist2018_dev	✓	✓	**3.07**	**0.183**	**0.137**
	nist2016_eval	✓	✓	4.18	0.258	0.206
	nist2019_eval	✓	✓	2.34	0.19	0.142

Table 2. Results of the prepared for the NIST SRE 21 CTS challenge obtained for NIST SRE 21 progress set

№	System name	Back-End	EER	DCF(0.05)	actDCF(0.05)
1	ECAPA-TDNN	Cosine	2.91	0.109	-
2	ResNet101	Cosine	2.75	0.097	-
3	ECAPA-TDNN + ResNet101	Cosine	2.59	0.091	-
4	ECAPA-TDNN + ResNet101	Cosine on cl-embeddings	2.71	0.085	-
5	ECAPA-TDNN+ResNet101+ResNet146+DTDNN	NBE	2.45	0.081	-
6	Fusion of 4 and 5	cosine & NBE	2.48	0.074	0.079
7	wav2vec-TDNN (clean)	Cosine on cl-embeddings	2.95	0.085	-
8	wav2vec-TDNN (aug)	Cosine on cl-embeddings	2.25	0.08	-
9	Fusion of 4 + 7 + 8	Cosine on cl-embeddings	2.42	0.072	0.083

for μ_2 and σ_2. A set of the *n* best scoring impostors are selected for each embedding pair when means and standard deviations are calculated.

CLLR loss function optimization was used to find scores scaling and shift parameters for appropriate scores calibration.

4. FINAL SUBMITTED SYSTEMS

The results of all considered single systems are presented in Table 1. One should note that using cl-embeddings and adaptive s-norm provide significant improvement. As the final submission system we used score and embedding level fusion of several single systems (see Table 2 for its performance results on the progress set). Systems 6 and 9 were calibrated using pooled NIST SRE 2016 and 2019 eval set.

5. REFERENCES

[1] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," Tech. Rep., 2019.

[2] Daniel Garcia-Romero, Greg Sell, and Alan McCree, "Magneto: X-vector magnitude estimation network plus offset for improved speaker recognition," in *Proc. Odyssey 2020 the speaker and language recognition workshop*, 2020, pp. 1–8.

[3] Aleksei Gusev, Vladimir Volokhov, Tseren Andzhukaev, Sergey Novoselov, Galina Lavrentyeva, Marina Volkova, Alice Gazizullina, Andrey Shulipa, Artem Gorlanov, Anastasia Avdeeva, et al., "Deep speaker embeddings for far-field speaker recognition on short utterances," *arXiv preprint arXiv:2002.06033*, 2020.

[4] Kong Aik Lee, Hitoshi Yamamoto, Koji Okabe, Qiongqiong Wang, Ling Guo, Takafumi Koshinaka, Jiacen Zhang, and Koichi Shinoda, "The nec-tt 2018 speaker verification system.," in *INTERSPEECH*, 2019, pp. 4355–4359.

[5] Omid Sadjadi, Craig Greenberg, Elliot Singer, Lisa Mason, Douglas Reynolds, et al., "Nist 2021 speaker recognition evaluation plan," 2021.

[6] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pretraining for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[8] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[9] Ya-Qi Yu and Wu-Jun Li, "Densely connected time delay neural network for speaker verification.," in *INTERSPEECH*, 2020, pp. 921–925.

[10] Sergey Novoselov, Andrey Shulipa, Ivan Kremnev, Alexandr Kozlov, and Vadim Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," *arXiv preprint arXiv:1804.10080*, 2018.

[11] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[12] Daniel Colibro, Claudio Vair, Emanuele Dalmasso, Kevin Farrell, Gennady Karvitsky, Sandro Cumani, and Pietro Laface, "Nuance–Politecnico di Torino's 2016 NIST speaker recognition evaluation system," in *INTERSPEECH 2017*, Stockholm, Sweden, August 2017, pp. 1338–1342.