ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414438

SYNAUG: SYNTHESIS-BASED DATA AUGMENTATION FOR TEXT-DEPENDENT SPEAKER VERIFICATION

Chenpeng Du, Bing Han, Shuai Wang, Yanmin Qian, Kai Yu

MoE Key Lab of Artificial Intelligence, AI Institute
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai,China
{duchenpeng, hanbing97, feixiang121976, yanminqian, kai.yu}@sjtu.edu.cn

ABSTRACT

Text-dependent speaker verification systems trained on large amount of labelled data exhibit remarkable performance. However, collecting the speech from a lot of speakers with target transcript is a lengthy and expensive process. In this work, we propose a synthesis based data augmentation method (SynAug) to expand the training set with more speakers and text-controlled synthesized speech. The performance of SynAug is evaluated on the RSR2015 dataset. Experimental results show that for i-vector framework, the proposed methods can boost the system performance significantly, especially for the low-resource condition where the amount of genuine speech is extremely limited. Moreover, combined with traditional data augmentation methods such as adding noises and reverberation, the systems could be further strengthened in extremely limited resource situation.

Index Terms— Data augmentation, Speech Synthesis, Text-dependent Speaker verification, i-vector

1. INTRODUCTION

Text-dependent speaker verification is the task of verifying whether the given speech belongs to the claimed speaker identity, in which the transcript is constrained to fixed lexical content. Both the traditional i-vector[1] systems and the deep learning based models, such as d-vector[2], j-vector[3] and x-vector[4], have been widely investigated. However, all these methods require sufficient amount of training data, while the collection of the text-dependent data is often very difficult and expensive.

To increase the amount and diversity of existing data, data augmentation is often applied as a pre-processing step when building deep learning models. For speaker verification tasks, different data augmentation methods are also proposed and analyzed in the literature. For example, by adding noises and reverberation to the clean audios, it's shown that the performance of x-vector systems can be significantly improved [4]. SpecAugment is a simple data augmentation method for speech recognition proposed in [5], which also shows its effectiveness for speaker verification tasks [6]. Sharing similar ideas, the random erasing strategy introduced in [7] is also proved to work well for the speaker verification tasks. Besides the augmentation for the front-end embedding extractors, researchers also investigated the application of generative adversarial network (GAN) and variational autoencoder (VAE) for the back-end PLDA augmentation [8, 9].

However, all the data augmentation approaches described above only provide variations on acoustic environment, which is only an aspect for the system robustness. Especially for the text-dependent tasks, the text variation should be explicitly considered. In this paper, inspired by the success of using synthetic speech in automatic speech recognition(ASR) [10, 11, 12], we propose a novel data augmentation approach, SynAug, that generates controlled speech of new speakers with a speech synthesis system for text-dependent speaker verification training. A main difference between this work and other TTS-based augmentation applied in ASR is that we can use additional text-independent speech as the reference to guide the synthesis. To the best of our knowledge, this is the first study to use synthetic speech for speaker verification training. The main advantages of SynAug are as follows,

1. The amount of speakers for training is increased, which enables the effective modelling of speaker identity information.

2. The generated speech share the same content with the target application, which is important for text dependent tasks.

The proposed method is examined on the RSR2015 dataset, the results obtained under the i-vector/PLDA framework exhibit the effectiveness of this approach.

Yanmin Qian is the corresponding author. This study was supported by the China NSFC projects No.62071288 and No.U1736202.

2. I-VECTOR

In the i-vector system, given an utterance, the speaker- and session-dependent supervector **M** is modeled as:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{x} \tag{1}$$

where **m** is the speaker and session-independent super-vector, **T** is a low rank matrix which captures speaker and session variability, referred to as the total variability matrix. The distribution of **x** is a standard normal distribution.

3. SYNAUG FOR TEXT-DEPENDENT SPEAKER VERIFICATION

3.1. Conditional FastSpeech2 based TTS system

Our TTS model in this work is based on FastSpeech2[13], which takes a phoneme sequence as input and the corresponding 320-dimensional mel-spectrogram as output. In this work, we need to synthesize speech for multiple speakers with variations, hence we use a condition extractor inspired by [14, 15, 16] to extract additional information other than the input phoneme from the reference speech, including speaker, speaking style, volume, speed, and etc. These information is expressed as the condition embedding **c** which is then broadcasted and added to the encoder output of FastSpeech2 for speech synthesis. The overall architecture of our TTS model is shown in Figure 1. In the training stage, the reference speech is exactly the target speech for training TTS. Therefore, the condition extractor is optimized to extract effective information in **c** for better reconstructing the mel-spectrogram. In the inference stage, we can randomly select a mel-spectrogram as the reference, and then obtain a synthetic speech corresponding to the given transcript with similar condition information to the reference.

The architecture of the condition extractor in this paper is similar to [14]. It contains 6 layers of 2D convolution with a kernel size of 3×3, each followed by a batch normalization layer and a ReLU activation function. A bidirectional GRU with a hidden size of 128 is designed after the above modules. The concatenated forward and backward states from the GRU layer is the output of the condition extractor, which is referred to as the condition embedding **c**.

3.2. The pipeline of SynAug

In this work, we assume that we have a limited text-dependent dataset $\mathcal{D}_{\mathrm{TD}}$ and a large text-independent dataset $\mathcal{D}_{\mathrm{TI}}$. Figure 2 demonstrates the pipeline of our data augmentation approach. We first train a TTS system on $\mathcal{D}_{\mathrm{TI}}$, and then synthesize new samples with the transcripts from $\mathcal{D}_{\mathrm{TD}}$ and speakers from $\mathcal{D}_{\mathrm{TI}}$. By sampling different speech of each speaker in $\mathcal{D}_{\mathrm{TI}}$ as the reference, we generate a synthetic text-dependent dataset $\mathcal{D}_{\mathrm{STD}}$, where each speaker has several different audios for each target transcript. For the i-vector systems, the UBM and PLDA are trained only on $\mathcal{D}_{\mathrm{TD}}$, while the i-vector extractor is trained on the pooled data of $\mathcal{D}_{\mathrm{TD}}$ and $\mathcal{D}_{\mathrm{STD}}$.

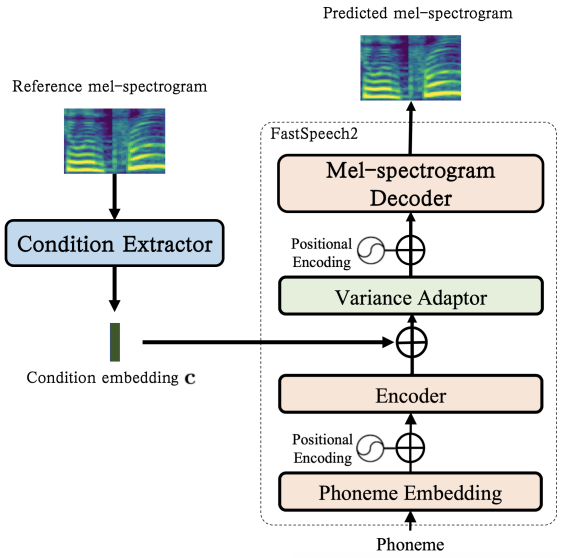

Fig. 1. Conditional FastSpeech2-based TTS architecture

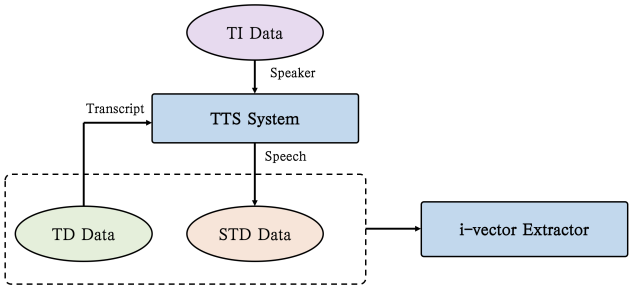

Fig. 2. The pipeline of SynAug. The TI, TD, and STD represents text-independent, text-dependent, and synthetic text-dependent data respectively. The TTS system is trained on TI data.

4. EXPERIMENT AND RESULTS

4.1. Dataset

The background set of RSR2015 [17] part1 corpus is used to train the speaker verification systems, which contains 97 speakers. The evaluation set from the same corpus is used to evaluate the proposed systems. The evaluation contains 1568008 trials, among which 19052 are target trials and 1548956 are impostor trials[1].

[1]Note that we removed all "impostor-wrong" trials since they are very easy to detect, leading to a very low EER, which makes the analysis non-intuitive

LibriTTS[18] is a large multi-speaker TTS dataset, whose training set is divided into two parts named "train-clean-460" and "train-other-500". We use train-clean-460 as the TTS training set, containing about 245 hours data. The speech is re-sampled to 16kHz for simplicity.

For the i-vector systems, we use 30-dimensional MFCC with a window size of 25ms and a frame shift of 10ms. The UBM has 512 Gaussian mixture components and the dimension of i-vector is set to 700.

In order to simulate the cases where different amount of TD data is available, we use 10, 20, 50 and all the 97 speakers in RSR2015 respectively in the experiments.

4.2. The necessity of speech sysnthesis

One trivial idea of data augmentation is to directly use the additional text-independent data $\mathcal{D}_{\mathrm{TI}}$ for training the i-vector extractor. Thus, it's necessary to show that directly introducing new speakers without constraining the speech content is not a good idea for text-dependent speaker verification. In this section, we will first show the importance of synthesizing speech with the desired content, and then analyze the impact of the synthesis's quality on the SV system.

First, we randomly select 200 speakers from the train-clean-460 dataset and directly use the corresponding text-independent data as the $\mathcal{D}_{\mathrm{TI}}$ for augmentation.

Then, we apply SynAug where we synthesize the 30 fixed transcripts in RSR2015 20 times for each of the same 200 speakers in $\mathcal{D}_{\mathrm{TI}}$. We use different utterances of the corresponding speakers as the references, in order to generate speech with diverse conditions. Figure 3 demonstrates an example of the mel-spectrograms of 4 instances generated by the TTS system with the same speaker and transcript. Despite their similarity, we can find obvious differences among the four mel-spectrograms, which shows the diversity of generated samples. We use Griffin-Lim algorithm [19] and WaveRNN [20] respectively to reconstruct the waveform from the predicted mel-spectrogram. The synthetic speech $\mathcal{D}_{\mathrm{STD}}$, together with the original text-dependent training data, is used for the i-vector training. We use only original data without augmentation in the PLDA stage.

Table 1. EER(%) on RSR2015 test set with different additional data. 10, 20, 50 and all the 97 genuine speakers' data are respectively mixed with the additional data for training.

Additional Data	Vocoder	Num spks used in RSR2015			
		10	20	50	97
None	-	9.26	2.52	1.07	0.71
TI	-	6.77	2.57	1.01	0.62
STD	Griffin-Lim	2.59	1.28	0.76	0.61
STD	WaveRNN	**2.18**	**1.14**	**0.71**	**0.61**

The results of the above systems are reported in Table 1. Generally, SynAug outperforms the text-independent data augmentation, which exhibits the importance of generating text-matched speech. We also find that the neural vocoder WaveRNN provides more improvements than Griffin-Lim algorithm because of the better voice quality of synthetic speech as mentioned in [20]. For example, SynAug with WaveRNN obtains a relative reduction of EER by 14.1% when all the 97 speakers in RSR2015 are available, and 76.5% when only 10 speakers in RSR2015 are available.

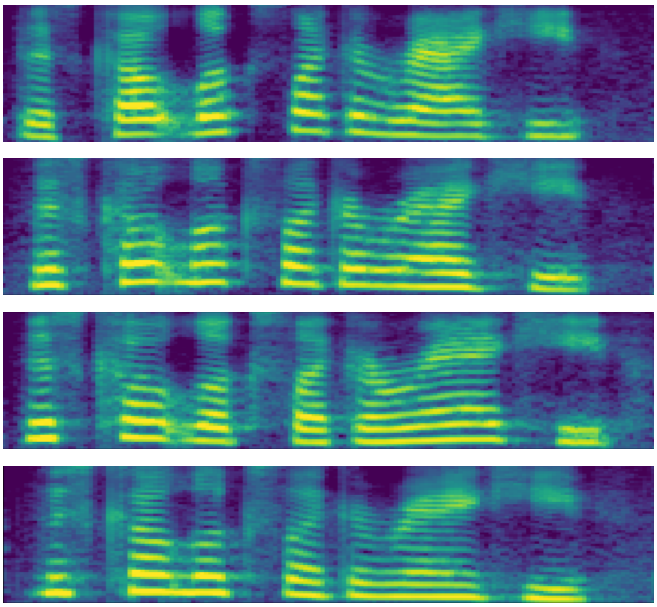

Fig. 3. Mel-spectrograms of 4 instances generated by the TTS system for the same speaker with the same transcript "this coat looks like a rag heap".

4.3. Impact of the SynAug scale

4.3.1. The number of speakers

In this section, we analyze how the number of speakers in the synthetic speech affects the performance of SynAug. We still simulate the 4 situations as described in 4.2 when different amount of TD data is available. The vocoder in the following experiments is the WaveRNN, which is better as discussed in Section 4.2. Then we randomly select 100, 200, 500 and 1000 speakers respectively from the train-clean-460 dataset and use them to synthesize the 30 fixed transcripts 20 times for SynAug.

The results are presented in Figure 4. We can find that the EER decreases when more speakers are used for SynAug. However, increasing the number of speakers in the synthetic speech cannot match increasing that in the real speech. For example, when we have 10 speakers in the real speech and 200 speakers in the synthetic speech, totaled 210 speakers, we get an EER of 2.18; but when we have 20 speakers in the real speech and 100 speakers in the synthetic speech, totaled

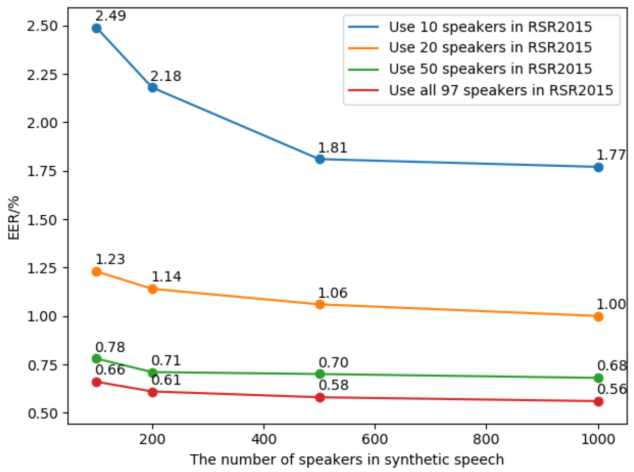

Fig. 4. EER(%) curves of i-vector systems on RSR2015 test set when different numbers of speakers are used in the synthetic speech.

120 speakers which is less than the previous combination, we get a lower EER of 1.23. This can partially be explained by the domain divergence between the speakers in RSR2015 and train-clean-460. Moreover, it is observed that increasing the number of speakers in the synthetic speech from 200 to 500 or 1000 obtains less improvement than increasing the number from 100 to 200. Therefore, we still use 200 speakers for SynAug in the following experiments for a balance between performance and computational cost.

4.3.2. The number of utterances for each speaker

In addition to the number of speakers, we also investigate the number of utterances for each speaker in the synthetic speech. Here, we synthesize the 30 transcripts 5, 10 and 20 times respectively for each speaker and observe the gains. As is shown in Table 2, when the number of utterances for each speaker grows, the EER decreases. This is in line with the common sense that more training data can provide better performance.

Table 2. EER(%) of i-vector systems on RSR2015 test set when different numbers of utterances are synthesized for each of the 200 speakers. 10, 20, 50 and all the 97 genuine speakers' data are respectively mixed with the synthetic data for training.

Num utts for each spk in synthetic speech	**Num spks used in RSR2015**			
	10	20	50	97
0	9.26	2.52	1.07	0.71
30 × 5	3.11	1.43	0.81	0.66
30 × 10	2.43	1.18	0.77	0.65
30 × 20	**2.18**	**1.14**	**0.71**	**0.61**

4.4. Combining the SynAug with adding noises and reverberation

We first present the results of adding noises and reverberation [4] for data augmentation. We follow the Kaldi Voxceleb recipe v2[21] and generate an augmented noisy copy of the original dataset. Both the original data and the generated noisy data are used for training the i-vector extractor. The results in Table 3 demonstrate that adding noises and reverberation can reduce EER when 10, 20, and 50 speakers are available in RSR2015.

Then, we combine the proposed SynAug method with adding noise and reverberation. Both the synthetic speech from TTS and the generated noisy speech mentioned above are used for i-vector training. We present the results in Table 3. The combination yields further gains in the low resource situations , compared with the systems using only TTS. However, when more speakers can be used in RSR2015, the combination brings no benefit. For example, when 10 speakers are available in RSR2015, the combination reduces the relative EER by 82.2% compared with the baseline that no data augmentation is applied, and 58.8% compared with the system that uses only noise and reverberation for data augmentation.

Table 3. EER(%) on RSR2015 test set when using different data augmentation approaches. "N. & R." represents noise and reverberation. 10, 20, 50 and all the 97 genuine speakers' data are respectively mixed with the augmented data for training.

Data Augmentation	**Num spks used in RSR2015**			
	10	20	50	97
None	9.26	2.52	1.07	0.71
N. & R.	4.00	1.90	1.01	0.74
SynAug	2.18	1.14	**0.71**	**0.61**
SynAug + N. & R.	**1.65**	**1.10**	0.75	0.66

5. CONCLUSION AND FUTURE WORK

In this work, we propose a novel synthesis based data augmentation method, SynAug, which generates speech of new speakers with a TTS system for text-dependent speaker verification training. By generating speech with controlled speech of new speakers, we show that SynAug can greatly benefit the text-dependent speaker verification systems, especially when the original training data is very limited. The experiments on RSR2015 dataset demonstrates that using 200 augmented speakers obtains a relative reduction of EER 14.1% when all the 97 speakers in RSR2015 are available, and 76.5% when only 10 speakers in RSR2015 are available. Moreover, combined with conventional augmentation methods such as adding noises and reverberation, the system performance could be further boosted.

6. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE ICASSP*, 2014, pp. 4052–4056.

[3] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Proc. ISCA Interspeech*, pp. 185–189.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE ICASSP*, 2018, pp. 5329–5333.

[5] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *CoRR*, vol. abs/1904.08779, 2019. [Online]. Available: http://arxiv.org/abs/1904.08779

[6] S. Wang, J. Rohdin, O. Plchot, L. Burget, K. Yu, and J. Černockỳ, "Investigation of specaugment for deep speaker embedding learning," in *Proc. IEEE ICASSP*. IEEE, 2020, pp. 7139–7143.

[7] V. Mingote, A. Miguel, D. Ribas, A. Ortega, and E. Lleida, "Knowledge distillation and random erasing data augmentation for text-dependent speaker verification," in *Proc. IEEE ICASSP*, 2020, pp. 6824–6828.

[8] Y. Yang, S. Wang, M. Sun, Y. Qian, and K. Yu, "Generative adversarial networks based x-vector augmentation for robust probabilistic linear discriminant analysis in speaker verification," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018, pp. 205–209.

[9] Z. Wu, S. Wang, Y. Qian, and K. Yu, "Data augmentation using variational autoencoder for embedding based speaker verification." in *Proc. ISCA Interspeech*, 2019, pp. 1163–1167.

[10] J. Li, R. Gadde, B. Ginsburg, and V. Lavrukhin, "Training neural speech recognition systems with synthetic speech augmentation," *CoRR*, vol. abs/1811.00707, 2018. [Online]. Available: http://arxiv.org/abs/1811.00707

[11] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. J. Moreno, Y. Wu, and Z. Wu, "Speech recognition with augmented synthesized speech," in *Proc. IEEE ASRU*, 2019, pp. 996–1002.

[12] C. Du and K. Yu, "Speaker augmentation for low resource speech recognition," in *Proc. IEEE ICASSP*, 2020, pp. 7719–7723.

[13] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *CoRR*, vol. abs/2006.04558, 2020. [Online]. Available: http://arxiv.org/abs/2006.04558

[14] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 5167–5176.

[15] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," in *Proc. ISCA Interspeech*, 2018, pp. 3067–3071.

[16] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proc. ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 4700–4709.

[17] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Rsr2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Proc. ISCA Interspeech*, 2012.

[18] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from librispeech for text-to-speech," in *Proc. ISCA Interspeech*, 2019, pp. 1526–1530.

[19] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[20] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 2415–2424.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE ASRU*, no. CONF. IEEE Signal Processing Society, 2011.