INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

The STC ASR System for the VOiCES from a Distance Challenge 2019

Ivan Medennikov[1,2], *Yuri Khokhlov*[1], *Aleksei Romanenko*[2], *Ivan Sorokin*[1], *Anton Mitrofanov*[1], *Vladimir Bataev*[1], *Andrei Andrusenko*[1], *Tatiana Prisyach*[1], *Mariya Korenevskaya*[1], *Oleg Petrov*[2,3], *Alexander Zatvornitskiy*[3]

[1] STC-innovations Ltd, St. Petersburg, Russia
[2] ITMO University, St. Petersburg, Russia
[3] Speech Technology Center Ltd, St. Petersburg, Russia

{medennikov,khokhlov,romanenko,sorokin,mitrofanov-aa,bataev,andrusenko, prisyach,korenevskaya,petrov-o}@speechpro.com, al.zatv@gmail.com

Abstract

This paper is a description of the Speech Technology Center (STC) automatic speech recognition (ASR) system for the "VOiCES from a Distance Challenge 2019". We participated in the Fixed condition of the ASR task, which means that the only training data available was an 80-hour subset of the LibriSpeech corpus. The main difficulty of the challenge is a mismatch between clean training data and distant noisy development/evaluation data. In order to tackle this, we applied room acoustics simulation and weighted prediction error (WPE) dereverberation. We also utilized well-known speaker adaptation using x-vector speaker embeddings, as well as novel room acoustics adaptation with R-vector room impulse response (RIR) embeddings. The system used a lattice-level combination of 6 acoustic models based on different pronunciation dictionaries and input features. N-best hypotheses were rescored with 3 neural network language models (NNLMs) trained on both words and sub-word units. NNLMs were also explored for out-of-vocabulary (OOV) words handling by means of artificial texts generation. The final system achieved Word Error Rate (WER) of 14.7% on the evaluation data, which is the best result in the challenge.

Index Terms: VOiCES19 Challenge, distant ASR, room simulation, speaker and room acoustics adaptation, R-vectors

1. Introduction

Distant Speech Recognition (DSR) is a highly important problem for a wide range of real-world applications, and at the same time, it is still far from being solved. A large number of initiatives aimed at studying the problem of DSR [1–4] has been organized in the last few years.

The VOiCES from a distance challenge 2019 is focused on benchmarking and further improving state-of-the-art technologies in the area of speaker recognition and ASR for far-field speech. For the ASR task, the organizers provided an 80h subset of the original LibriSpeech [5] corpus for models training. Development and evaluation data used during the challenge are parts of the recently released Voices Obscured in Complex Environmental Settings (VOiCES) corpus [6]. A detailed description of the provided data can be found in [6, 7]. It should be noted that the training data were recorded on a close microphone in quiet conditions, while the VOiCES corpus consists of utterances recorded on distant microphones in noisy environments. Thus, the mismatch between the training and testing conditions is the main challenge.

There are numerous approaches developed for improving the quality of DSR systems. First, data augmentation techniques such as room acoustics simulation [8] are extremely useful. Next, dereverberation and denoising methods based on signal processing (e.g. WPE [9]) and deep learning (e.g. denoising wavenet [10], approaches based on generative adversarial networks (GAN) [11]) can reduce the interference effects and clean the acoustic signal from external distortion sources. Auditory model based features such as Gabor filterbanks [12] and gammatone filterbanks [13] are often used in DSR because of their robustness to difficult acoustic environments. Acoustic distortion compensation such as Vector Taylor Series [14] can be used to improve the ASR system quality. Beamforming approaches should also be mentioned, however, they do not make sense in the VOiCES challenge, as the data are single-channel.

This paper provides a description of the STC ASR system for the Fixed condition of the VOiCES challenge. Figure 1 shows a scheme of the system. First, Kaldi cleanup procedure [15] was applied to the training data. Then, in order to generate large-scale simulated data, we applied room acoustics simulation in a way similar to one described in [8]. After that, recordings were dereverberated using the WPE algorithm [9]. The resulting enhanced data were used to obtain x-vector speaker embeddings [16, 17], R-vector RIR embeddings [18], and three types of features: 80-dimensional log Mel filterbank features computed by Kaldi [19] (fbank) and Librosa [20] (libfb), and gammatone filterbank [13] features of size 64 (gtf). Per-utterance normalized features, along with their deltas and delta-deltas, were combined with x- and R-vector embeddings. The resulting feature maps were used as inputs of LF-MMI [21] acoustic model based on Convolutional Neural Network (CNN) [22] followed by Factorized Time-Delay Neural Network (TDNN-F) [23]. Additional step of state-level Minimum Bayes Risk (sMBR) [24] training was performed. Lattice-level combination of 6 systems based on 4 pronunciation dictionaries was applied. Finally, we performed 1000-best list rescoring using a combination of 3 NNLMs operating on words, Byte Pair Encoding (BPE) sub-words [25] and Morfessor sub-words [26] respectively. We also submitted a single system without NNLM rescoring, which uses 530k vocabulary language model (LM) built on artificially generated texts.

The rest of the paper is organized as follows. Section 2 describes a system chosen as a baseline. Data preparation is covered in Section 3. Section 4 presents acoustic modeling, including adaptation to the speaker and room acoustics. Section 5 describes advanced language models as well as OOV handling. Results of our submission are presented in Section 6. Finally, Section 7 concludes the paper and discusses the results.

http://dx.doi.org/10.21437/Interspeech.2019-1574

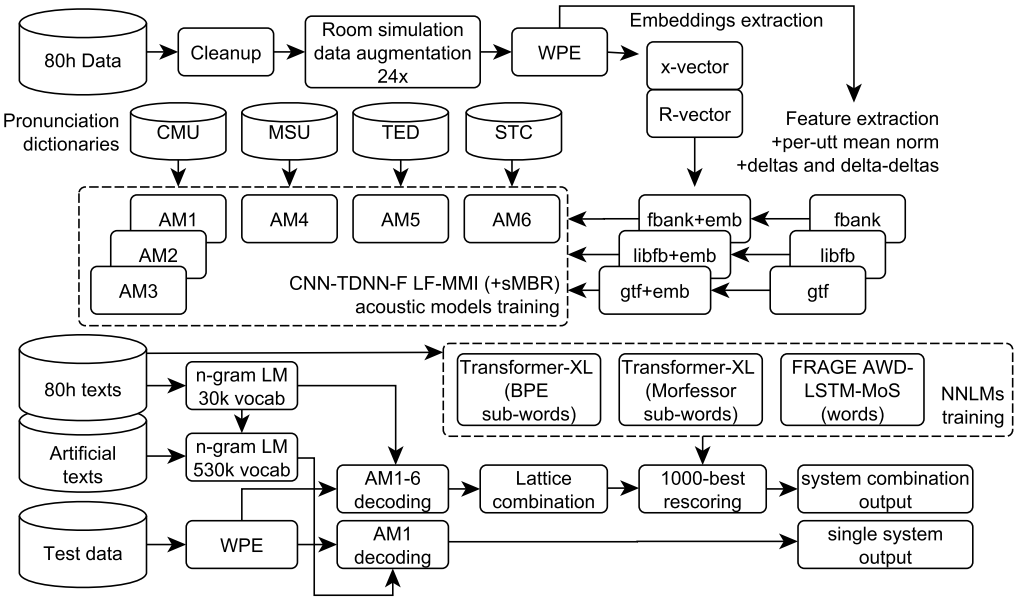

Figure 1: *Schematic diagram of the proposed system*

2. Baseline system

The 80h training set was used to build the baseline system. We followed a slightly modified Kaldi [19] LibriSpeech recipe (s5) which involves flat-start acoustic models training. Baseline language model (LM) was a 3-gram LM with a 30k vocabulary. Phoneme transcriptions were taken from the CMUdict version 0.7b[1]. GMM-HMM model *tri4* was used for the cleanup procedure [15] and for building a lexicon with pronunciation probabilities [27]. After that, GMM-HMM model *tri4b_cleaned* was trained, and the alignment from this model was used for further acoustic models training. For neural network acoustic models training, we removed speed perturbation step of the recipe. I-vector extractor was trained using standard Kaldi procedure. Final acoustic model for our baseline system was Factorized Time Delay Neural Network (TDNN-F) [23] trained with LF-MMI criterion [21] on cleaned data and fbank features with i-vectors. Performance of the baseline system is presented in Table 1.

Table 1: *Performance of the baseline on the development set*

Acoustic model	WER(%)
tri4b (no pron. prob.)	80.00
tri4b	77.77
tri4b_cleaned	76.02
TDNN-F LF-MMI	**68.78**

3. Data augmentation and dereverberation

In order to reduce the mismatch between the training and development audio, as well as to increase the amount of training data, we investigated room acoustics simulation data augmentation [8]. The image method [28] was used for RIRs generation in time-[2] [29] and frequency-domain[3] [30]. For both setups, the number of reflections was limited to 8. The parameters of simulated environments (SNR, T60, distance from the target source to the microphone, etc.) were similar to those used in [8].

Along with the RIR simulation, the original data were modified by adding various noises. The noise set was a combination of MUSAN [31] (music part only), noises from the AURORA 2 corpus [32], QUT-NOISE [33] and about 20h of television noise selected similarly to [6]. The number of added noises per utterance varied from 0 to 3. The original training data were augmented 24 times, resulting in about 1894 hours of simulated training data. Note that on each augmentation step we selected randomly 10% of utterances that were not augmented.

Time- and frequency-domain room simulation approaches were compared on 8-fold simulated training data (one-third of the full amount). As shown in Table 2, the frequency-domain room acoustics simulation performed better in terms of WER, so we used this approach in further experiments.

Table 2: *WER(%) on the development set with time- and frequency-domain room simulation*

Baseline	Time-domain	Frequency-domain
68.78	24.56	**23.04**

We also applied the open-source implementation[4] [34] of the WPE [9] algorithm to reduce reverberation effects. As can be seen from Table 3, WPE dereverberation significantly reduces WER on the development set. WPE applied to simulated training audio resulted in an additional WER reduction. Therefore, for building final acoustic models we applied WPE to both

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] https://github.com/ehabets/RIR-Generator

[3] https://github.com/sunits/rir_simulator_python

[4] https://github.com/fgnt/nara_wpe

development and simulated training data.

Table 3: *Comparison of WPE applied to training and development data in terms of WER(%)*

	dev original	**dev WPE**
train original	23.04	22.26
train WPE	23.55	**21.80**

4. Acoustic modeling

4.1. Acoustic model architectures

In addition to the baseline 15-layer TDNN-F acoustic model, we investigated the following architectures:

- CNN-TDNN-F: 7-layer CNN followed by 9-layer TDNN-F.
- TDNN-LSTMP: a mix of 10 TDNN-F layers and 3 Long Short-Term Memory layers with projections (LSTMP).
- BLSTMP-F: 3-layer bidirectional LSTMP followed by a linear layer with self-orthogonal constraint.

Experiments with AM architectures were performed with 8-fold room simulation augmented training data. WPE dereverberation was not applied in these experiments.

Table 4: *Comparison of acoustic model architectures*

Acoustic model	**WER(%)**
TDNN-F	23.04
CNN-TDNN-F	**20.53**
TDNN-LSTMP	25.20
BLSTMP-F	27.42

Table 4 shows that CNN-TDNN-F architecture significantly outperforms the others, so we chose it for all further experiments.

4.2. Acoustic model adaptation using speaker embeddings and RIR embeddings

Acoustic model speaker adaptation using speaker embeddings, e.g. i-vectors, is a widely used technique [35]. Our CHiME5 experience [36] revealed that speaker adaptation is extremely useful not only for telephone and microphone speech recognition tasks but for distant ASR as well. Based on this experience, we decided to apply modern x-vectors speaker embeddings [16, 17] instead of i-vectors for speaker adaptation. The extractor of 512-dimensional x-vectors was trained on a full WPE-processed 24-fold simulated data (1894h) using slightly modified Kaldi recipe described in [16].

On the other hand, we supposed that room acoustics adaptation could be even more helpful than speaker adaptation in distant ASR. Following this idea, we exploited the Kaldi x-vectors framework for building RIR classifier and extracting 512-dimensional RIR embeddings called R-vectors. More specifically, 30k RIRs were classified using 8 simulated utterances as examples for each class. Details on room acoustics adaptation can be found in a companion paper [18] which is focused on a deep investigation of the proposed method.

As x-vectors training and extracting procedure requires Speech Activity Detector (SAD), we used ASR-derived per-frame SAD marks. For the training data, senone forced alignment obtained on the original clean data was used. For development and evaluation data, we converted ASR results to SAD marks using Kaldi *lattice-best-path* tool. It should be noted that the ASR system applied for obtaining SAD marks was built without the use of any external speech data.

X-vectors and R-vectors, in general, do not contain spatially contiguous patterns, therefore they can not be applied directly in CNN acoustic models. In order to overcome this, we transformed embedding input into 5 additional feature maps using an affine layer. Results with speaker and room acoustics adaptation, as well as their combination, are given in Table 5 (full 24-fold simulated training data, WPE for both training and development data is applied). It can be seen that room acoustics adaptation provided a more significant improvement than speaker adaptation. Moreover, these approaches are complementary.

Table 5: *Acoustic model speaker and room acoustics adaptation results on the development set*

Adaptation	**WER(%)**
no	19.40
x-vector	17.54
R-vector	16.99
x-vector + R-vector	**16.47**

4.3. Final acoustic models

Final acoustic models were trained on WPE-processed 24-fold simulated data with some minor improvements in the training procedure. Specifically, 3 sets of input features (fbank, libfb and gtf) were used; first and second order derivatives were added as additional feature maps (provided WER improvement up to 0.5%); backstitch training [37] with scale 0.15 was applied for each minibatch (WER improvement about 0.3%); most of the models were fine-tuned using sMBR sequence-discriminative criterion [24] (WER reduction up to 0.4%). In addition to the baseline CMU dictionary, we also utilized MSU[5], TED-LIUM [38], and our proprietary (STC) English dictionaries. As shown in Table 6, different models perform close, and lattice-level combination reduces WER significantly.

Table 6: *Final AMs and results on the development set*

#	**Dictionary**	**Features**	**Loss**	**WER(%)**
1	CMU	fbank	sMBR	15.31
2	CMU	libfb	sMBR	15.36
3	CMU	gtf	sMBR	15.62
4	MSU	fbank	LFMMI	15.64
5	STC	fbank	sMBR	15.44
6	TED	fbank	sMBR	15.53
combination 1+2+3+4+5+6				**14.25**

[5] https://www.isip.piconepress.com/projects/switchboard/releases/sw-ms98-dict.text

5. Advanced language modeling

5.1. Neural network based language models

In order to improve the 1-best recognition hypotheses, we explored 3 types of NNLMs. Along with recurrent neural network language model (RNN-LM) integrated with the Kaldi toolkit [39], we also investigated Transformer-XL [40] and FRequency AGnostic word Embedding (FRAGE) with ASGD Weight-Dropped (AWD) Long Short-Term Memory (LSTM) Mixture of Softmaxes (MoS) [41], which are the current state-of-the-art for large and medium size vocabulary language modeling tasks respectively. Initial RNN-LM, Transformer-XL, and FRAGE with AWD-LSTM-MoS were trained on word-level training set transcriptions. However, only 4 MB of training texts were available, so we assumed that the data sparsity problem can impact NNLMs training. In order to overcome this, two sets of sub-word units (7663 sub-word units generated by Morfessor[6] [26] and 9135 BPE sub-word units [25]) were used for training additional Transformer-XL LMs.

Table 7: *Rescoring results for different language models on the development set*

#	Language model	WER(%)
0	n-gram	15.31
1	Kaldi RNN-LM (words)	14.03
2	Transformer-XL (words)	13.30
3	Transformer-XL (Morfessor sub-words)	13.63
4	Transformer-XL (BPE sub-words)	13.43
5	FRAGE (words)	13.32
	combination 3+4+5	**12.86**

RNN-LM was applied to rescore lattices [42], while the other NNLMs rescored 1000-best recognition hypotheses. Rescoring results for system #1 from Table 6 with different NNLMs are presented in Table 7. It can be seen that Transformer-XL LMs and FRAGE with AWD-LSTM-MoS LM demonstrated the best results in terms of WER. Word-level Transformer-XL and FRAGE with AWD-LSTM-MoS performed almost equally. We included the last one to the combination because its architecture significantly differs from the Transformer-XL based sub-word NNLMs.

5.2. OOV handling

It is well known that high OOV rate causes ASR system performance degradation. Regarding the baseline 30k vocabulary, development set contained 2255 unique OOV words (5657 hits, which is 3% of the total number). In order to handle this, we explored the augmentation of LM training data using artificially generated texts [43]. Several text generators based on character-level NNLMs were trained on the whole amount of training set transcriptions, namely char-rnn[7] [44], Textgenrnn[8] (original and reversed versions), and TrellisNet [45]. The total amount of artificially generated texts was about 464MB, containing more than 3 million new unique words. We trained n-gram LM on these texts and interpolated it with the baseline LM. The vocabulary of the final LM was limited to 500k most frequent generated words, along with 30k original words. This allowed to cover 3133 OOV hits out of 5657 (55.38%) on the development set and resulted in WER reduction from 15.31% to 14.19% for system #1 from Table 6. Unfortunately, due to time limitations and a large amount of required computations, this approach was included only to our single-system submission without NNLM rescoring.

[6] https://github.com/aalto-speech/morfessor

[7] https://github.com/karpathy/char-rnn

[8] https://github.com/minimaxir/textgenrnn

6. Final submission and results

Table 8 presents our 2 submissions and their results on the development and evaluation data. The first one is an extended 530k vocabulary single system without NNLM rescoring. The second one is a lattice-level combination of 6 systems with the original 30k vocabulary, followed by 1000-best rescoring with a combination of 3 NNLMs. It should be noted that the small difference between these numbers and ones presented in Section 5 is due to the scoring procedure (here, the official GLM file is used in order to normalize between different orthographical conventions).

Table 8: *Final submission details*

System	Vocab	NNLM	WER dev	WER eval
single	530k	no	14.0	17.8
combination	30k	yes	12.4	14.7

7. Conclusion and discussion

We described the STC_ASR team contribution to the VOiCES challenge. The mismatch between training and testing conditions was overcome by the room acoustics simulation. The proposed R-vectors based room acoustics adaptation approach turned to be very useful and, moreover, to be complementary with x-vectors based speaker adaptation.

A lot of speech enhancement approaches (GAN for dereverberation [11], wavenet for denoising [10], enhancer based on ideal ratio masks (IRM)[9]) were investigated, but only the WPE algorithm was applied in the final submissions. The enhancers mentioned above provided some improvement for models trained on clean training data, however, they increased WER for models trained on a large-scale simulated data. The joint training of the IRM enhancer and acoustic model on a simulated data gave about 2% absolute WER reduction, but this improvement vanished after applying x-vectors and R-vectors. That is why this approach was not included in the final system.

Speech synthesis based data augmentation was explored but without any improvement. We think that speech synthesis models were not good enough because of a too small amount of training data. We also expected that the end-to-end systems would be helpful in OOV words handling, but the lack of data led to the poor performance of such models.

8. Acknowledgements

We would like to thank STC_SID team which participated in the speaker recognition task of the VOiCES Challenge [46] for their help with building x-vector based speaker embeddings extractor. We also thank our colleague Ilya Kalinovskiy for valuable discussion on data augmentation approaches.

This work was partially financially supported by the Government of the Russian Federation (Grant 08-08).

[9] https://github.com/funcwj/setk

9. References

[1] M. Harper, "The automatic speech recogition in reverberant environments (ASpIRE) challenge," in *ASRU*, 2015, pp. 547–554.

[2] K. Kinoshita *et al.*, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, 2013.

[3] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, *The CHiME Challenges: Robust Speech Recognition in Everyday Environments*, 2017, pp. 327–344.

[4] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *INTERSPEECH*, 2018.

[5] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[6] C. Richey, M. A. Barrios, Z. Armstrong *et al.*, "Voices obscured in complex environmental settings (VOiCES) corpus," in *INTERSPEECH*, 2018, pp. 1566–1570.

[7] M. K. Nandwana, J. van Hout, M. McLaren *et al.*, "The VOiCES from a distance challenge 2019 evaluation plan," *arXiv:1902.10828 [eess.AS]*, 2019.

[8] C. Kim, A. Misra, K. K. Chin *et al.*, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *INTERSPEECH*, 2017, pp. 379–383.

[9] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[10] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," *ICASSP*, pp. 5069–5073, 2018.

[11] K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, "Investigating generative adversarial networks based speech dereverberation for robust speech recognition," in *INTERSPEECH*, 2018.

[12] M. Kleinschmidt, "Localized spectro-temporal features for automatic speech recognition," in *INTERSPEECH*, 2003.

[13] H. K. Maganti and M. Matassoni, "An auditory based modulation spectral feature for reverberant speech recognition," in *INTERSPEECH*, 2010.

[14] M. Korenevsky, "Phase term modeling for enhanced feature-space VTS," *Speech Communication*, vol. 89, pp. 84–91, 2017.

[15] V. Peddinti, V. Manohar, Y. Wang, D. Povey, and S. Khudanpur, "Far-field ASR without parallel data," in *INTERSPEECH*, 2016, pp. 1996–2000.

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

[17] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," in *Odyssey*, 2018, pp. 378–385.

[18] Y. Khokhlov, A. Zatvornitskiy, I. Medennikov *et al.*, "R-vectors: New technique for adaptation to room acoustics," in *INTERSPEECH (accepted)*, 2019.

[19] D. Povey, A. Ghoshal, G. Boulianne *et al.*, "The kaldi speech recognition toolkit," *ASRU*, 2011.

[20] B. McFee *et al.*, "librosa: Audio and music signal analysis in python," in *Python in Science Conference*, 2015, pp. 18–24.

[21] D. Povey, V. Peddinti, D. Galvez *et al.*, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH*, pp. 2751–2755.

[22] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *INTERSPEECH*, 2016.

[23] D. Povey, G. Cheng, Y. Wang *et al.*, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018, pp. 3743–3747.

[24] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, 2013.

[25] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv:1508.07909*, 2015.

[26] S. Virpioja, P. Smit, S.-A. Grönroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," Tech. Rep., 2013.

[27] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for ASR," in *INTERSPEECH*, 2015.

[28] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Acoustical Society of America Journal*, vol. 65, no. 4, pp. 943–950, 1979.

[29] S. McGovern, "The image-source reverberation model in an n-dimensional space," in *DAFx-11*, 2011.

[30] D. R. Campbell, K. J. Palomäki, and G. J. Brown, "A MATLAB simulation of shoebox room acoustics for use in research and teaching," *Computing and Information Systems Journal*, vol. 9, 01 2005.

[31] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv:1510.08484*, 2015.

[32] D. Pearce and H.-G. Hirsch, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000, pp. 29–32.

[33] D. B. Dean, A. Kanagasundaram, H. Ghaemmaghami, M. H. Rahman, and S. Sridharan, "The QUT-NOISE-SRE protocol for the evaluation of noisy speaker recognition," in *INTERSPEECH*, 2015, pp. 3456–3460.

[34] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *13. ITG Fachtagung Sprachkommunikation (ITG 2018)*, Oct 2018.

[35] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2013, pp. 55–59.

[36] I. Medennikov, I. Sorokin, A. Romanenko *et al.*, "The STC system for the CHiME 2018 challenge," in *CHiME5 Workshop*, 2018.

[37] Y. Wang, V. Peddinti, H. Xu *et al.*, "Backstitch: Counteracting finite-sample bias via negative steps," in *INTERSPEECH*, 2017.

[38] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an Automatic Speech Recognition dedicated corpus," in *LREC*, 2012.

[39] H. Xu, K. Li, Y. Wang *et al.*, "Neural network language modeling with letter-based features and importance sampling," *ICASSP*, pp. 6109–6113, 2018.

[40] Z. Dai, Z. Yang, Y. Yang *et al.*, "Transformer-XL: Attentive language models beyond a fixed-length context," *arXiv:1901.02860*, 2019.

[41] C. Gong, D. He, X. Tan *et al.*, "FRAGE: Frequency-agnostic word representation," in *NeurIPS*, 2018.

[42] H. Xu *et al.*, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," *ICASSP*, pp. 5929–5933, 2018.

[43] Y. Khokhlov, I. Medennikov, A. Romanenko *et al.*, "The STC keyword search system for OpenKWS 2016 evaluation," in *INTERSPEECH*, 2017, pp. 3602–3606.

[44] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," 2015. [Online]. Available: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

[45] S. Bai, J. Z. Kolter, and V. Koltun, "Trellis networks for sequence modeling," in *ICLR*, 2019.

[46] S. Novoselov, A. Gusev, A. Ivanov *et al.*, "STC speaker recognition systems for the VOiCES from a distance challenge," in *INTERSPEECH (accepted)*, 2019.