

TOWARDS DATA-EFFICIENT MODELING FOR WAKE WORD SPOTTING

Yixin Gao, Yuriy Mishchenko, Anish Shah, Spyros Matsoukas, Shiv Vitaladevuni

Alexa, Amazon.com Inc.
101 Main St, Cambridge, MA 02142, USA

ABSTRACT

Wake word (WW) spotting is challenging in far-field not only because of the interference in signal transmission but also the complexity in acoustic environment. Traditional WW model training requires large amount of in-domain WW-specific data with substantial human annotations. This prevents the model building in the situation of lacking such data. In this paper we present data-efficient solutions to address the challenges in WW modeling, such as domain-mismatch, noisy conditions, limited annotation, etc. Our proposed system is composed of a multi-condition training pipeline with stratified data augmentation, which improves the model robustness to a variety of predefined acoustic conditions, together with a semi-supervised learning pipeline to extract the WW and adversarial examples from untranscribed speech corpus. Starting from only 10 hours of domain-mismatched WW audio, we are able to enlarge and enrich the training dataset by 20-100 times to capture the complexity in acoustic environments. Our experiments on real user data show that the proposed solutions can achieve comparable performance of a production-grade model by saving 97% of the amount of WW-specific data to collect and 86% of the bandwidth for annotation.

Index Terms— wake word spotting, far-field, multi-condition training, semi-supervised learning

1. INTRODUCTION

Wake word (WW) is the gatekeeper that enables end users to interact with the cloud-based voice assistants. To ensure good customer experience, a WW spotter needs to be highly sensitive to detect the WW from low signal-to-noise ratio (SNR) speech, and robust to minimize false alarms, thereby protecting the users' privacy. WW spotters are widely deployed on the voice-enabled smart home devices such as Amazon Echo, Google Home, Apple HomePod, etc. These devices are often installed in complicated acoustic conditions and used from a distance, therefore the speech signal arrived at the device is heavily interfered and attenuated by the dynamical environment, resulting in many difficulties in far-field WW detection.

Tremendous efforts have been made in recent years for improving far-field keyword spotting or WW detection. Most of them focused on investigating novel model architectures [1, 2, 3, 4, 5, 6, 7, 8] and improving the efficiency of training [9, 10, 11, 12, 13]. These efforts implicitly assume that the transcribed in-domain WW-specific training data is sufficient for building WW models. The training data used in these works ranges from 200 hours to tens of thousands of hours, and most of them were collected from far-field devices in real use and transcribed by human annotators to ensure that the richness of the background is accurately captured by the training data.

For a new WW, collecting and transcribing such large size of speech corpus from matched domain, for example far-field, is time-consuming and cost-prohibitive, which can be a roadblock

for building a spotter for the new WW. Therefore how to develop data-efficient solutions for WW model building becomes the key question. In this paper, we developed solutions to build far-field WW spotter using as little as 10 hrs ($\sim 10K$ utterances) of the WW training data that was collected from close-talk microphones (CTM) in a quiet acoustic environment. In many situations CTM data is more readily available, and compared with far-field data is much easier to obtain, for example by using crowd-sourcing [14]. Our approach uses a stratified data augmentation for multi-condition training, together with a semi-supervised learning (SSL) employing an automatic speech recognition (ASR) model to extract WW and adversarial examples from untranscribed speech corpus.

In literature data augmentation has been used in ASR [15, 16, 17, 11], however traditional approaches simplistically apply arbitrary distortions, or random clipping, or transformation of audio in single pass indiscriminately. Such strategy is much less effective when working with extremely limited amount of data. On the contrary, in this paper we first define specific uses cases, such as clean close-talk, noisy close-talk, clean far-field, and noisy far-field, and further explore how these cases, generated from limited number of initial clean utterances, should be used in the training dataset for the best performance of WW model in real world.

There are many open questions for using untranscribed data for WW spotting, for example, how to accurately extract WW samples from inaccurate results of modern ASR recognition; how to balance the mined dataset to avoid bias; how to automatically scale up to arbitrary new WWs, new language, etc. For discriminative training, how to define the hard examples and how to efficiently obtain them is not known. Traditional approach of blindly sampling from non-WW audio [1, 2, 9] does not optimize the model discrimination. In this paper, we answer these questions by the proposed a semi-supervised learning framework equipped with a mechanism called "lexicon filter" to effectively extract confusing examples for a given wake word.

Our experiments on real user data show that the proposed methods enable us to build WW models that matches the performance of a production-grade WW models trained with several hundreds of hours of transcribed WW data collected from real far-field scenarios. To the best of our knowledge, it is the first work in literature that is focused on how to use such little training data for building production-quality far-field wake word spotters. In addition, we benchmark the performance gain from each method to provide practical references for a variety of use cases for building new WW spotting models such as CTM data only; clean far-field data only; untranscribed speech corpus only, etc

2. SYSTEM OVERVIEW

The proposed training system for WW spotter is shown in the Figure 1. The system is composed of two data pipelines: multi-condition and semi-supervised learning, and a multi-task training pipeline.

The multi-condition data pipeline applies a stratified data aug-

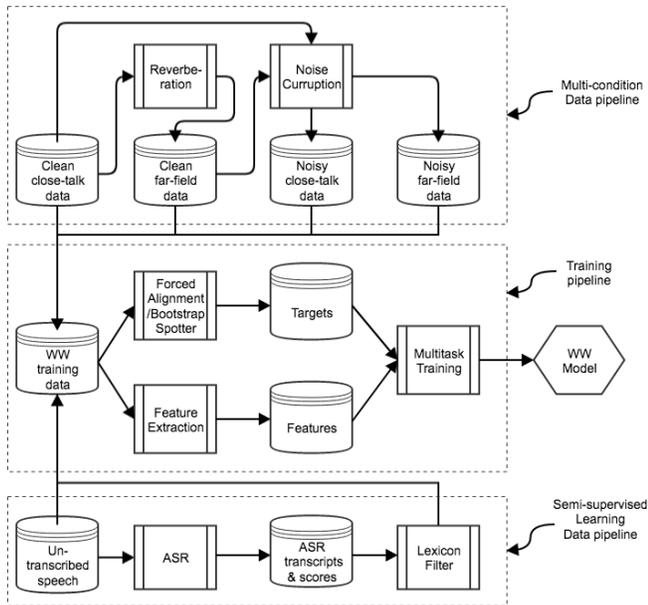


Fig. 1: The proposed WW training system.

mentation to a small set of clean close-talk speech data. Specifically, the CTM data was first convolved with audio recordings from a library of room impulse response (RIR) to mimic the reverberated speech received by a far-field device. Then both the CTM and reverberated speech are corrupted by additive noise profiles at different levels of SNR, generalizing the CTM data to become a representation of the audio signal in typical household use cases, i.e. mixed conditions of clean near-field, noisy near-field, clean far-field and noisy far-field environments.

The semi-supervised learning data pipeline uses an ASR model to obtain the automatic transcripts with confidence scores, and is equipped with a mechanism called “lexicon filter” to mine utterances with WW and adversarial examples from untranscribed speech corpus. The Lexicon filter uses the Levenshtein distance as a measurement of pronunciation proximity to filter automatic transcripts. Each data pipeline can be plug into the training system separately, and they can be combined together to complement each other. We will discuss the effects of each method in section 5.

3. MULTI-CONDITION TRAINING

Multi-condition training (MCT) [18], i.e., training with data from heterogeneous sources and diversified conditions, enables a model to learn higher level features that are more invariant to noise, and was shown to be effective in robust ASR [19, 20]. In this paper we use data augmentation to simulate training data associated with conditions that are weakly represented in the original CTM data, such as far-field and/or noisy environment, and employ the mixed-condition training approach [19] to train a DNN WW spotter. Our data augmentation strategy for multi-condition training contains two components: reverberation and noise corruption.

3.1. Reverberation: from close-talk to far-field

The far-field speech can be obtained by the close-talk speech convoluted with the room-impulse response filters:

$$x_R(t) = x_S(t) * r(t) \quad (1)$$

where r is the room impulse response, x_S is the source close-talk signal and x_R is the reverberated signal, respectively. The image-source method, as presented originally in [21], can be used to precisely simulate the RIRs for small rooms with known measurements. However the computation grows exponentially with the order of reflections thus preventing it from practical usage.

In this paper we applied a proprietary library of RIR filters [11] to mimic the far-field audios. The RIR filters library is collected through real devices set up in real rooms simulating household living rooms by playing audio signal such as chirps at different locations. By varying the locations of the devices or the sound sources, the RIR library approximated the effects of playing the audio at a random locations in the room.

3.2. Corruption: improving the noise robustness

Since our close-talk data is from clean acoustic environment, the reverberated data is also acoustically clean, hence it does not reflect well the real noisy conditions. We employ two internally collected noise data libraries: household noise and music/movie audio, to improve the performance in both general household and media playing conditions. The simulated noisy far-field data can be achieved by

$$x_A(t) = x(t) + \alpha \cdot n(t) + \beta \cdot m(t) \quad (2)$$

where $x \in x_R \cup x_S$ is the clean original CTM or reverberated signal, x_A denotes the corrupted signal, n and m are noise or music interferences, respectively. α and β are the scaling factors to control the ratio of the two noise sources and they are coupled with the target SNR, which is randomly drawn from a normal distribution with mean 10.0 dB and standard-deviation 3.0 dB.

3.3. Mixed-condition training

We use a combination of the above data augmentation methods to create a mixed-condition training set of clean, clean+reverb, clean+noise, clean+reverb+noise speech. Any labels, such as the DNN targets, of the augmented data are the same as the labels of original data. We pool the mixed-condition data together and train a single DNN model, which is also called mixed-conditional model or multi-style model [19].

4. SEMI-SUPERVISED LEARNING

The augmented dataset is limited in the diversity of the source, such as the size of the vocabulary, the population of speakers etc. A general speech corpus used for training ASR models, which usually contains thousands of hours of audio and human-prepared parallel transcripts, incorporates much greater variety in sources, nevertheless the majority of such speech corpus is not relevant to WW. There may exist large corpora of untranscribed audio data with significant amounts of WW-relevant data, but they are hard to use because of lack of human annotations.

In this paper we develop a simple but efficient semi-supervised learning strategy to mine useful WW-training data from large untranscribed general-speech corpora. We employ a general far-field ASR model to automatically transcribe the unlabeled speech corpus, then select WW-specific utterances and add them to our training set. The DNN targets are obtained by force aligning the speech to the automatic transcripts. Since the ASR transcript may not be accurate, only the utterances with WW confidence score higher than a threshold θ_P are selected. It should be noted that, in the utterances mined

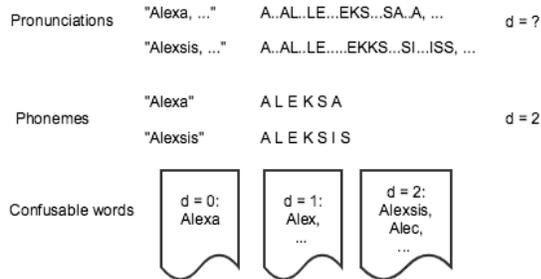


Fig. 2: Lexicon filter for “Alexa”.

here the WW may appear in the middle of the utterance, therefore the phonetic context may be different from the utterances initiated by the WW. In practice, our experiments show that such context mismatch have unobservable effects on WW spotting.

4.1. Lexicon filter

In machine learning overemphasizing hard examples during training commonly leads to better model discrimination and generalization. For a particular WW, ideally a hard example is a combination of syllables that pronounces similarly to the WW. However accurately calculating the similarity of pronunciations of the words in two sentences is computationally intractable. Here we propose a tool called “lexicon filter”, which calculates the Levenshtein distances [22] with the target WW over the phoneme representations of words provided by the lexicon from the ASR model as an approximation of the pronunciation similarity to define the set of confusing words for the WW. Since the word dictionary in ASR model is usually very large, and the majority of the words rarely appear in a speech corpus, constraining the search within the most frequent words would save a lot of computations. This set is used to filter the ASR transcripts for utterances containing the confusing words. The utterances containing a confusing word with confidence score higher than a threshold θ_N are added to training set as the negative examples. Without loss of generality we use the popular wake word “Alexa” to explain the concept of the lexicon filter in Fig. 2.

4.2. Loss function

The SSL loss for DNN training is

$$L_{SSL} = \sum_{\substack{x \in \mathbf{u} \\ \mathbf{u} \in \mathcal{P} \cup \mathcal{N}}} \left(\mathbb{1}(\mathbf{u} \in \mathcal{P}) y \log \frac{1}{q(x)} + (1 - y) \log \frac{1}{1 - q(x)} \right) \quad (3)$$

where \mathcal{P} denotes the set of WW (positive) examples ($d = 0$) and \mathcal{N} denotes the set of adversarial (negative) examples ($d \geq 1$), $x \in \mathbf{u}$ denotes the context-dependent feature frame in an utterance \mathbf{u} , $q(x)$ denotes the DNN output posterior probability for x , and $y \in \mathcal{Y}$ is the corresponding framewise DNN target obtained by forced alignment using the ASR transcripts, i.e.

$$\mathbf{y} = fa(\mathbf{u}, ASR(\mathbf{u})) \quad (4)$$

where \mathbf{u} denote the utterances in the untranscribed speech corpus, fa denotes the forced alignment operator[23], respectively.

The thresholds θ_P and θ_N can be tuned to balance the ratio of positive and negative examples in the training set. In our experiments we choose both θ_P and θ_N as 0.5. In addition, for a six-phoneme wake word, we found the lexicon filter threshold on the Levenshtein distance $d = 1, 2$ produces sufficient negative examples for augmenting the training with confusing cases..

5. EXPERIMENTS

5.1. Model architecture

The wake word spotting system used in this paper is composed of a compressed feed-forward deep neural network (DNN), a posterior smoothing and a peak detection modules, as shown in Fig. 3. The input audio signal to the WW spotter is resampled at 16KHz and a 20-bin log filterbank energy (LFBE) feature is computed every 10 msec with a 25 msec observation window. The LFBE features from a context window of 31 frames (20 on the left and 10 on the right) are stacked into a 620-dimensional vector and fed into the DNN. The DNN is composed of three non-linear fully-connected layers with 400 nodes, and before each non-linear layer a linear bottleneck layer with 87 nodes is inserted by decomposing the weighting matrix into two smaller matrices, similar to [10, 4, 24]. The output layer is a two-dimensional softmax, which can be easily extend to multiple WWs. For decoding, the posteriors corresponding to the WW are first smoothed by a windowed moving average, where the window size is approximately the same as the average duration of the WW, then the thresholding and peak-detection algorithm is applied to infer both the location and duration of a WW detection.

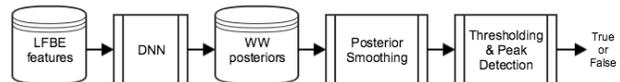


Fig. 3: The WW spotter.

5.2. Data sets

We choose “Amazon” as our new wake word for experiments. A total of 9487 utterances beginning with the chosen WW were recorded from a close-talk microphone in quiet room, denoted as “CTM10K” dataset. The average duration of the utterances is 3.86 secs and the total duration of the dataset is 10.18 hrs. For comparison, a baseline production-grade WW model was trained using 375 hours of transcribed speech data collected from far-field devices in real use cases.

The evaluation dataset contains a composition of more than 300 hours of audio collected from far-field, mid-field and near-field devices in real use cases from various acoustic conditions. The dataset is sufficiently large to show strong statistical significance.

5.3. Experimental setup

We use the GPU-based distributed trainer as described in [25] for DNN training. The models are trained using transfer-learning paradigm where the weights of the DNN are initialized by an ASR acoustic model of the same architecture and size trained with ASR senone targets. During training a multi-task learning framework similar to [9, 4, 7, 8, 11] is used where the loss is a weighted sum of WW loss and ASR loss. The ASR branch regularizes the DNN training and is helpful especially when training dataset is small.

The evaluation metrics in this paper are false reject rate (FRR), which is 1 minus recall, and false alarm rate (FAR), which is a normalized number of false accepts. Absolute values of FAR in this paper have been anonymized for confidentiality reasons. The range of used FAR corresponds to the range normally used for production keyword spotting models. We report the two metrics in a detection-error-tradeoff (DET) curve as we tune the decision thresholds for each model. The lower the DET curve, the better the performance.

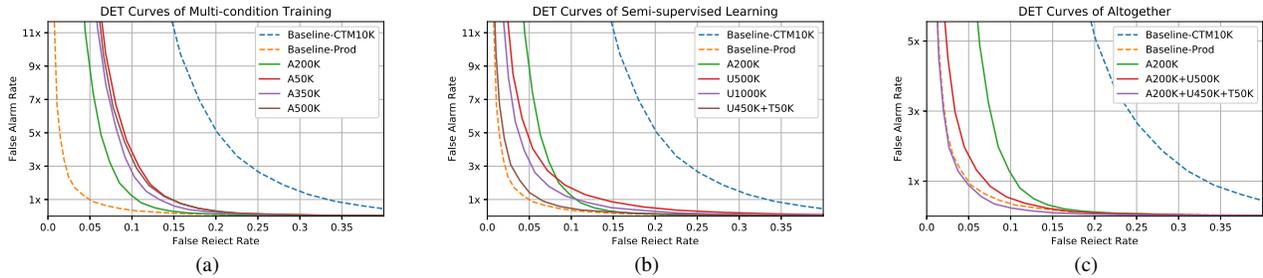


Fig. 4: DET curves with (a) multi-condition training sets of different sizes; (b) untranscribed and partial transcribed data; (c) multi-condition training and semi-supervised learning combined.

5.4. Results of multi-condition training

Our data augmentation strategy defines the training sets to constitute conditions denoted as (CTM, CTM+R, CTM+N and CTM+RN), where CTM, R, N, RN denote the original close-talk microphone data, reverberation, noise corruption, reverberation followed by corruption, respectively. We sweep the size of the training set by increasing the portion of clean CTM data to as much as 10% and mixing with equal amount of the three augmented datasets. The augmentation used in our experiments are detailed in Table 1

Table 1: Details of the multi-condition training sets.

size	CTM	CTM+R	CTM+N	CTM+RN
50K	10K	14K	14K	14K
200K	20K	60K	60K	60K
350K	50K	100K	100K	100K
500K	50K	150K	150K	150K

Fig. 4 (a) shows the DET curves of the augmented training sets (prefixed with “A”) of size 50K, 200K, 350K and 500K respectively, compared to the two baselines: “Baseline-CTM10K” (blue) and “Baseline-Prod”(orange). The DET curves show that the multi-condition training boosts the performance significantly. In our experiments 20x augmentation (A200K) achieves the maximum gain, which suggests that overwhelming augmentation may hurt the performance. There remains a gap between the best multi-condition DET curve (A200K) and the Baseline-Prod DET curve, indicating some mismatch exists between the simulated data and the real-world.

5.5. Results of semi-supervised learning

We use the the semi-supervised learning (SSL) strategy described in section 4 to construct two datasets (prefixed with “U”) of 500K and 1000K total utterances from the untranscribed in-house production speech corpus by employing a production-grade ASR model similar to [26]. Both datasets contain equal portion of utterances containing the WW and the confusing words as described in 4.1.

Fig. 4 (b) show the DET curves of the WW model trained with those datasets. The untranscribed datasets also boost the WW detection performance significantly, and U1000K achieves more gain than U500K. Note that a gap still exists between the DET curves of U1000K and Baseline-Prod, which suggests that only untranscribed data is not enough to achieve production performance. We further transcribed 10% of the 500K utterances to study the effect of human input and created the dataset called “U450K+T50K”. The model

trained with this dataset outperforms the models with untranscribed-only data or the augmented-only data, and the DET curve is very close to the DET curve of the Baseline-Prod.

5.6. Results of MCT and SSL combined

Having observed that both the multi-condition training (MCT) and semi-supervised learning (SSL) approaches work better at different regions, we hypothesize that they supplement each other to achieve better performance if combined together. Therefore we take the best augmented dataset (A200K), and add to the untranscribed or the partially-transcribed dataset, denoted as A200K+U500K and A200K+T50K+U450K, respectively, and the result DET curves are shown in Fig. 4 (c). We can observe that the A200K+U500K DET curve becomes much closer to the Baseline-Prod DET curve. Moreover, with 50K utterances transcribed, the A200K+T50K+U450K DET curve is further improved and performs slightly better than Baseline-Prod model. This observation indicates that each approach (multi-condition training, semi-supervised learning, transcription) has own strength and limitation, and can supplement each other to achieve better than production-quality performance.

6. CONCLUSION

In this paper we proposed a series of methods based on multi-condition training and semi-supervised learning to model new wake word spotter using as little as 10 hours (~ 10 K) clean anechoic recordings. By employing a stratified data augmentation to prepare 200K multi-condition data, added to 500K untranscribed utterances selected from a far-field speech corpus, together with transcribing 50K utterances, we are able to match a production-quality baseline. Comparatively, we saved 97% (1-10/375) in data collection and 86% (1-50/375) in transcription.

Our work provides solutions to many different scenarios emerging during developing models for new WW, and the performance benchmarks (eg. @FRR=0.05): (1) If only the close-talk recordings are available, the multi-condition training brings us within 9x FAR from the Baseline-Prod; (2) if an untranscribed far-field speech corpus is available, the semi-supervised learning alone can bring us within 5x FAR; (3) if both augmented data and far-field corpus are available, combining the two can further bring us within 2x FAR; (4) if transcribed far-field data is present, or researchers have the ability to transcribe data, adding up to 50K transcribed utterances together with 10 times of untranscribed speech allow us to build models that can match/outperform the production quality.

7. REFERENCES

- [1] Guoguo Chen, Carolina Parada, and Georg Heigold, “Small-footprint keyword spotting using deep neural networks,” in *IEEE ICASSP*, 2014.
- [2] Tara N. Sainath and Carolina Parada, “Convolutional neural networks for small-footprint keyword spotting,” in *INTER-SPEECH*, 2015.
- [3] Yanzhang He, Rohit Prabhavalkar, Kanishka Rao, Wei Li, Anton Bakhtin, and Ian McGraw, “Streaming small-footprint keyword spotting using sequence-to-sequence models,” in *ASRU*, 2017.
- [4] Ming Sun, David Snyder, Yixin Gao, Varun K. Nagaraja, Mike Rodehorst, Sankaran Panchapagesan, Nikko Strom, Spyros Matsoukas, and Shiv Vitaladevuni, “Compressed time delay neural network for small-footprint keyword spotting,” in *INTERSPEECH*, 2017.
- [5] Kenichi Kumatani, Sankaran Panchapagesan, Minhua Wu, Minjae Kim, Nikko Strom, Gautam Tiwari, and Arindam Mandal, “Direct modeling of raw audio with dnns for wake word detection,” in *ASRU*, 2017, pp. 252–257.
- [6] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, “Attention-based end-to-end models for small-footprint keyword spotting,” in *INTERSPEECH*, 2018.
- [7] Jinxi Guo, Kenichi Kumatani, Ming Sun, Minhua Wu, Anirudh Raju, Nikko Strom, and Arindam Mandal, “Time-delayed bottleneck highway networks using a dft feature for keyword spotting,” in *IEEE ICASSP*, 2018.
- [8] Minhua Wu, Sankaran Panchapagesan, Ming Sun, Jiacheng Gu, Ryan Thomas, Shiv Vitaladevuni, Bjorn Hoffmeister, and Arindam Mandal, “Monophone-based background modeling for two-stage on-device wake word detection,” in *IEEE ICASSP*, 2018.
- [9] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Bjorn Hoffmeister, and Shiv Vitaladevuni, “Multi-task learning and weighted cross-entropy for dnn-based keyword spotting,” in *INTERSPEECH*, 2016.
- [10] George Tucker, Minhua Wu, Ming Sun, Sankaran Panchapagesan, Gengshen Fu, and Shiv Vitaladevuni, “Model compression applied to small-footprint keyword spotting,” in *INTER-SPEECH*, 2016.
- [11] Anirudh Raju, Sankaran Panchapagesan, Xing Liu, Arindam Mandal, and Nikko Strom, “Data augmentation for robust keyword spotting under playback interference,” in *arXiv:1808.00563*, 2018.
- [12] Jinyu Li, Rui Zhao, Zhuo Chen, Changliang Liu, Xiong Xiao, Guoli Ye, and Yifan Gong, “Developing far-field speaker system via teacher-student learning,” in *INTERSPEECH*, 2018.
- [13] Yangbin Chen, Tom Ko, Lifeng Shang, Xiao Chen, Xin Jiang, and Qing Li, “Meta learning for few-shot keyword spotting,” in *arXiv preprint arXiv:1812.10233*, 2018.
- [14] Pete Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” *CoRR*, vol. abs/1804.03209, 2018.
- [15] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *INTERSPEECH*, 2015.
- [16] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019.
- [17] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *ICASSP*, 2017.
- [18] Michael L. Seltzer, Dong Yu, and Yongqiang Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *ICASSP*, 2013.
- [19] Yan Huang, Malcolm Slaney, Yifan Gong, and Mike Seltzer, “Towards better performance with heterogeneous training data in acoustic modeling using deep neural networks,” in *Interspeech*, 2014.
- [20] Matthew Gibson, Christian Plahl, Puming Zhan, and Gary Cook, “Multi-condition deep neural network training,” in *ESSV2018 - 29th Conference on Electronic Speech Signal Processing*, 2018.
- [21] Jont B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *The Journal of the Acoustical Society of America*, no. 65, pp. 943–950, 1979.
- [22] Gonzalo Navarro, “A guided tour to approximate string matching,” *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, Mar. 2001.
- [23] Frederick Jelinek, “Continuous speech recognition by statistical methods,” in *Proceedings of the IEEE 64*, 1976, pp. 532–556.
- [24] Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran, “Low-rank matrix factorization for deep neural network training with high-dimensional output targets,” in *ICASSP*, 2013.
- [25] Nikko Strom, “Scalable distributed dnn training using commodity gpu cloud computing,” in *INTERSPEECH*, 2015.
- [26] Roland Maas, Ariya Rastrow, Kyle Goehner, Gautam Tiwari, Shaun Joseph, and Bjorn Hoffmeister, “Domain-specific utterance end-point detection for speech recognition,” in *INTER-SPEECH*, 2017.