INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

The 2020 Personalized Voice Trigger Challenge: Open Datasets, Evaluation Metrics, Baseline System and Results

Yan Jia[1,4], *Xingming Wang*[1,3], *Xiaoyi Qin*[1,3],*Yinping Zhang*[2],*Xuyang Wang*[2],*Junjie Wang*[2], *Dong Zhang*[4], *Ming Li*[1,3]*

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]AI Lab of Lenovo Research, Beijing, China
[3]School of Computer Science, Wuhan University, Wuhan, China
[4]School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

ming.li369@dukekunshan.edu.cn

Abstract

The 2020 Personalized Voice Trigger Challenge (PVTC2020) addresses two different research problems in a unified setup: joint wake-up word detection with speaker verification on close-talking single microphone data and far-field multi-channel microphone array data. Specially, the second task poses an additional cross-channel matching challenge on top of the far-field condition. To simulate the real-life application scenario, the enrollment utterances are recorded from close-talking cell-phone only, while the test utterances are recorded from both the close-talking cell-phone and the far-field microphone arrays. This paper introduces our challenge setup and the released database as well as the evaluation metrics. In addition, we present a sequential two stage end-to-end neural network baseline system trained with the proposed database for speaker-dependent wake-up word detection. Results show that state-of-the-art personalized voice trigger methods are still based on the two stage design, however, this benchmark database could also be used to evaluate multi-task joint learning methods. The official website [1], the open-source baseline system[2] and results[3] of submitted systems have been released.

Index Terms: open source database, wake-up word detection, speaker verification, joint learning

1. Introduction

Speaker dependent voice trigger and wake-up word detection are gaining popularity among speech researchers and developers. It has been deployed in many real-life applications. With the contribution of deep learning, the performance of wake-up word detection and speaker recognition systems have improved remarkably in both close-talking and far-field scenarios. The demand for authentication based on voice technologies, including keyword spotting (KWS) and text-dependent speaker verification (TDSV), is growing rapidly for personalized voice trigger devices. Generally, the KWS aims to detect a predefined keyword or a set of keywords in a continuous audio stream. Recently, End-to-End Deep Neural Networks (DNNs) are applied to KWS and show that DNN based methods perform well compared with Hidden Markov Model(HMM) based wake-up systems[1]. Since then, more complex network structures have been adopted to build end-to-end KWS systems, including Convolutional Neural Networks[2], Recurrent Neural Networks[3, 4], etc. On the other hand, with the success of deep learning in the speaker verification (SV) field[5] and the demand for personalized trigger in smart home devices, the TDSV task has attracted much attention of speaker verification researchers.

The 2020 Personalized Voice Trigger Challenge (PVTC2020), which aims at providing a common platform for the research community to advance the state-of-the-art techniques in this field. The PVTC2020 challenge is focused on the speaker dependent wake-up word detection. We release a database named XIAO-LE[4] containing recordings of wake-up words under the smart home scenario in this challenge. Besides, we also provide an open source two-stage speaker dependent KWS baseline system. When the KWS system triggers, we compare the trigger audio with the reference model created during the registration process and use another threshold to determine whether the sound that triggers the detector may be the wake-up word uttered by the registered user.

This rest of the paper is organized as follows. The details of XIAO-LE data is introduced in Section 2, and in section 3, the design features and evaluation metrics of the challenge are presented. Section 4 describes the personalized KWS baseline system. Section 5 discusses the experimental results. Conclusion is provided in section 6.

2. The XIAO-LE Database

The XIAO-LE database is provided by the AI Lab of Lenovo Research. It contains 658,995 utterances with 612 hours in total. The database covers 550 speakers and a wide range of channels from close-talking microphones to multiple far-field microphone arrays. It can be used for far-field wake-up word recognition, far-field speaker verification, and speech enhancement.

The average duration of all utterances is around 3.8 seconds. During the recording process, recording devices, including two cell phones (16kHz, 16bit) and four microphone arrays (with 4 or 6 channels per array, 16kHz, 16bit), were set in a room designed as the real smart home environment [5].

For the data collected by microphone arrays, each audio file has 4- or 6-channel signals, while for the data collected by cell

*Corresponding author

[1] https://www.pvtc2020.org

[2] https://github.com/lenovo-voice/THE-2020-PERSONALIZED-VOICE-TRIGGER-CHALLENGE-BASELINE-SYSTEM

[3] https://www.pvtc2020.org/leaderboard.html

[4] https://www.pvtc2020.org/dataDescription.html

[5] https://www.pvtc2020.org/faq.html

http://dx.doi.org/10.21437/Interspeech.2021-602

phones, each utterance only has two channels signal. Recording devices and their corresponding distance information are shown in table 1.

Table 1: *Distance of different recording devices.*

Devices_id	Device and distance
id1	Cell phone, 0.2m
id2	Cell phone, 0.8m
id3	Microphone array, 1m
id4	Microphone array, 3m
id5	Microphone array, 3m
id6	Microphone array, 5m

3. Challenge Setup

3.1. Task Settings

Based on the XIAO-LE database, we have divided it into a training set, a development set, and two evaluation sets. Specifically, 300 speakers are selected for training, and 50 speakers are used as the development set. The rest speaker of the database is used for evaluation. The challenge provides two tracks for the participants, and the second task poses a cross-channel challenge.

3.1.1. Task 1: Joint wake-up word detection with speaker verification on close-talking data

Only data collected by cell phones in the evaluation set from 100 speakers is adopted for performance evaluation in the first task. The evaluation set was separated into enrollment data and testing data. For each target speaker, the positive testing samples have 'xiao le, xiao le' as a part of the speech, and it is indeed uttered from the target speaker. There might be some background noises or even other speakers' voices before the target speaker says 'xiao le, xiao le'. However, all utterances considered as positive samples have 'xiao le, xiao le' uttered by the target speaker at the end of the speech in this case. Negative samples do not contain speech segments with the content 'xiao le, xiao le' uttered by the target speaker. Note that utterances without 'xiao le, xiao le' or with 'xiao le, xiao le' that are not uttered by the target speaker are both considered as negative samples.

3.1.2. Task 2: Joint wake-up word detection with speaker verification on far-field multi-channel microphone array data

For the second task, we adopt data from another 100 speakers in the evaluation set with no overlapping with the one in task 1. The evaluation data and the trials are constructed in the same way as task 1. The only difference is that the testing data only includes multi-channel synchronized audio data collected by one of those far-field microphone arrays. Similar to task 1, 'xiao le, xiao le' is always at the end of the sentence for all positive samples. In contrast, negative samples do not contain 'xiao le, xiao le' or have speech segments 'xiao le, xiao le' that are not uttered by the target speaker. The details about the trial design are shown in Table 2.

3.2. Design of trial files

For speaker verification, participants can use up to three audios for enrollment for each speaker. The trial we provided contains three selected enrollment audios, one test audio, and the label which denotes whether the trial is positive or negative. In task 1, the utterances collected by cell phone in 0.2m are selected as the enrollment data, and those utterances collected by cell phone in both distances with 0.2m and 0.8m are used as the testing data. In task2, the utterances collected by cell phone in 0.2m distance are selected as the enrollment data, while utterances collected by multi-channel far-field microphone arrays are used for testing. We describe different scenarios in the trial construction with more details in Table 2. The distribution of different trials are shown in Table 3.

3.3. Evaluation Metrics

In this challenge, we provide a leaderboard ranked by the metric $score_{wake\_up}$. The speaker dependent KWS performance of our baseline system, as well as systems submitted by participants in the challenge, are measured by this metric. The $score_{wake\_up}$ is calculated from the miss rate and the false alarm (FA) rate according to the following equation,

$$score_{wake\_up} = Miss + alpha * FA \quad (1)$$

Miss represents the proportion of errors in all positive label samples, and FA refers to the rate of errors in all negative label samples. The $alpha$ constant is set as 19, which is calculated by assuming that the ratio of positive to all samples is 0.05.

In addition, the real-time factor ($F_{real-time}$) is also evaluated as an auxiliary metric, which is calculated as the overall processing time of the evaluation trials on an Intel Core i5 core clocked at 2.6 GHz or similar processors divided by the total duration of all the testing samples. That is calculated as follows:

$$F_{real-time} = T_{process}(s)/T_{total\_test}(s) \quad (2)$$

$T_{process}(s)$ is the overall time cost of processing all the evaluation data in seconds, and $T_{total\_test}(s)$ is the total duration of the testing audios. In task2, multi-channel data will be considered as single-channel data when calculating T_{total_test}. Besides, extracting the speaker embedding or features from the enrollment data is not counted in $T_{process}(s)$. $F_{real-time}$ is a mandatory self-reported metric. Each submission is considered as a valid submission only when the corresponding self-reported real-time factor is lower than the given threshold of 0.25.

4. The Baseline Methods

4.1. LSTM-based KWS system

This section presents our KWS baseline system, which is modified from the CNN-based KWS system in [6]. Our baseline system consists of three modules:(i) a feature extraction module, (ii) a stacked LSTM neural network and (iii) a confidence calculation module.

The feature extraction module converts the audio signals into acoustic features. 80 dimensional log-mel filterbank features are extracted for speech frames with 25ms long and 10ms shift. Then we apply a segmental window with 40 frames to generate training samples that contain enough context information of sub-word as the input of the model.

Our backbone network is constructed with a two-layer stacked LSTM structure, followed by an average pooling layer and a final linear projection layer. For all LSTM layers, the hidden dimension is set to 128. A fully connected layer and a final softmax activation layer are applied as the back-end prediction module to obtain the subword occurrence probability of predefined keywords.

Table 2: *Structure of the Trial files. Noting that, other text independent segments denote speech segments other than 'xiao le, xiao le'*

Includes 'xiao le, xiao le'	'xiao le, xiao le'part is from the target speaker	Includes other text independent segments from non-target speakers before 'xiao le, xiao le'	Includes other text independent segments from the target speaker	Label
yes	yes	no	no	positive
yes	yes	no	yes	positive
yes	yes	yes	no	positive
yes	no	no	no	negative
yes	no	no	yes	negative
yes	no	yes	no	negative
no	n/a	n/a	yes	negative
no	n/a	n/a	no	negative

Table 3: *Details about the development and test set*

		utterances	positive	negative	enrollment
Development	Task1	24.9k	3.6k	23.5k	1.6k
	Task2	50.1k	4.6k	48.5k	1.6k
Evaluation	Task1	159.2k	19.7k	148.1k	3.1k
	Task2	201.7k	28.8k	190.5k	3.1k

In the posterior handling module, while the acoustic feature sequence is projected to a posterior probability sequence of keywords by the neural network, we adopt the method proposed in [7, 8] to make detection decisions. In this approach, we apply a sliding window with the length of T_{conf} frames to compute detection scores and denote the input acoustic features in a window as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_{T_{conf}}\}$. $\boldsymbol{w} = \{w_1, w_2 \cdots w_M\}$ represents the sub-words sequence of pre-defined keywords. Then the output confidence score $h(\boldsymbol{X})$ is computed by equation 3,

$$h(\boldsymbol{X}) = \left[\max_{1 \leq t_1 < \cdots \leq T_{conf}} \prod_{i=1}^{M} p_{w_i}(\boldsymbol{x}_{ti}) \right]^{\frac{1}{M}}, \quad (3)$$

where $p_{w_i}(\boldsymbol{x}_{ti})$ refers to the network output of t^{th} frame at subword w_i.

This method is suitable for the real-time situation. The system triggers whenever the confidence score is higher than the pre-defined threshold.

4.2. Speaker verification system

The training process of the speaker verification baseline system is modified from the framework in [14]. The whole architecture contains a front-end feature extractor, an encoding layer and a back-end classifier. We used ResNet34 [15] with SE-block [16] as the feature extractor. The attentive statistics pooling(ASP) [17] is adopt as the encoding layer. The ASP layer uses an attention mechanism to give different weights to different frames and generates a weighted average and a weighted standard deviation at the same time, which can effectively capture longer-term speaker feature variations. The AM-Softmax [18] was set as the back-end classifier in the system.

4.3. Speaker dependent KWS system

Our baseline system consists of a wake-up system and a speaker verification system described above. As shown in Figure 1, we designe a two-stage system that responds whenever the target speaker says the trigger phrase. When the KWS system triggers, the speaker verification system starts to decide whether the voice that triggers the detector is likely to be from the enrolled user. During enrollment stage, the average vector of the three utterances' speaker embedding is saved as the enrollment speaker embedding vector.

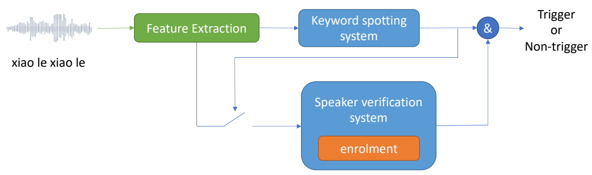

Figure 1: *Framework of the baseline system.*

We compare any possible new 'xiao le, xiao le' utterance with the stored templates as follows. The first stage detector produces timing information used to convert the acoustic feature sequence into a fixed-length vector. A separate, specially trained speaker verification network transforms this vector into a speaker embedding. We compare the cosine between the testing embedding and the reference template created during enrollment with another threshold to decide whether the sound that triggers the wake-up word detector is likely to be the one from the enrolled speaker. This process can help reduce the cases where the device is triggered by 'xiao le, xiao le' spoken by another person and reduces the rate at which other confusing trigger phrases.

4.4. Experiment setup

4.4.1. Keyword spotting system

We determine target word labels by force-alignment with an LVCSR system. For keyword 'xiao le, xiao le', the ending time of the first 'xiao', the first 'le', and the second 'xiao' are found out and we center its on a window of 40 frames. 80 dims log fbank is adopted as our input acoustic features. The KWS system is trained with cross-entropy loss. Stochastic gradient descent with Nesterov momentum is selected as the optimizer. The learning rate is first initialized as 0.01 and decreases by a factor of 0.1 whenever the model reaches a training loss plateau. We train the KWS model for 100 epochs with a batch size of 128 and employ early stopping when the training loss is not decreasing. In the evaluation period, we use a sliding window of 150 frames to compute the confidence score.

Table 4: *The methods and results of top performing systems*

Team Name	**Data**	**KWS**	**SV System**	**Note**	**Result**		
					$F_{real-time}$	Task1	Task2
The Xiaomi-NPU System	PVTC+openslr Data Augmentation	TDNN[9] CNN-TDNNF[10]	x-vector[5] ResNet34	Two-stage KWS+Two-stage SV Location Augmentation MMD loss+CORAL loss	0.206	0.075	0.084
The NPU System	PVTC+openslr Data Augmentation SpecAugment	MDTC[11]	ResNet34[11]	Location Augmentation Binary Cross Entropy Location Estimation	0.120	0.080	0.091
Mobvoi beyond AI System	PVTC+openslr SpecAugment	TDNN TCN	ResNet34	Focal loss	0.172	0.113	-
Seedland Corp AI System	PVTC+openslr Data Augmentation	MobileNet[12] LSTM	Resnet34SEV2[13]	-	-	0.209	0.200
Baseline V2	PVTC+openslr	LSTM	Resnet34	-	0.201	0.37	0.31
Baseline V1	PVTC+openslr	LSTM	Resnet34	-	0.203	0.75	0.78

4.4.2. Speaker verification system

According to the experiments in [19], the strategy of transfer learning performs well in the far-field text-dependent speaker verification tasks. Therefore, we select the data from SLR38, SLR47[20], SLR62, SLR82[21], SLR85[22] on openslr[6] as the pre-training data. After that, we carry out fine-tuning on the XIAO-LE database. Fine-tuning schemes are divided into two types: the first is to use all the utterances of the pre-training database to construct a text-independent speaker verification system as a pre-training model; the second is to use only the database of XIAO-LE to fine-tune the pre-training model and get the target text-dependent system. We adopt online data augmentation to improve the robustness of the speaker verification system[23]. We use the MUSAN [24] and the RIRs-NOISES [25] as the noise sources. The signal-to-noise-ratio(SNR) was set between 0 to 20 dB while pre-training and 0 to 15 dB while fine-tuning. For pre-training, we also use stochastic gradient descent as the optimizer. The initial learning rate is set as 0.01 and decreases by 0.1 per 20 epochs. The pre-trained model is trained for 50 epochs with a batch size of 256. For fine-tuning, the initial learning rate is set to 0.001 and the number of training epochs is set to 20.

The threshold of the speaker verification system was determined by the development set. Two ad-hoc ways to determine the threshold have been used in our baseline system. The first method is using the threshold of EER (Equal Error Rate) as the baseline version 1 system. The second is using the mean threshold of EER and minDCF[26], which greatly improved in the development set as baseline version 2 system.

5. Results

5.1. Results

We received sixteen systems for the first task, and three systems for the second task. For two tasks, only the top three systems are included in Table 4, the results of all participating teams with valid submissions can be found on the leaderboard on the Challenge website. All the experiments are evaluated on an Intel I5 series CPU clocked at 2.5 GHz. Table 4 presents the methods and results of the submitted systems as well as the baseline method.

From Table 4, we can observe that the recordings of task 2 are all far-field scene, and the performance of the model on task 2 decreases significantly compared to task 1. Second, the methods plays an important role in determining the final score.

As for submitted systems, using a more complex neural network in KWS can achieve better results than the baseline system. The parameters of the four submitted KWS networks are more than those of the baseline system. The Xiaomi-NPU system achieves the best performance on the both task1 and task2, because they adopt the complex system structure of two-stage KWS models and two-stage speaker verification models. All submitted systems use data augmentation methods to expand the original training set of the challenge, which is critical for our system to generalize to the development set as well as the evaluation set. For the development set and the evaluation set, the keyword always appears at the end of the utterance, and keyword always appears at the beginning of the utterance in original training data. The two top performing systems use ASR force alignment information to augment new data where the key word may appear at random positions in the utterances to make the KWS model more robust, which significantly improves the performance on the evaluation set.

6. Conclusions

In this paper, we introduce the setup of the 2020 Personalized Voice Trigger Challenge (PVTC2020) and describe the datasets, tracks, rules, baseline systems, evaluation metrics and results of the challenge. Results show that state-of-the-art personalized voice trigger methods are still based on the two stage design and data augmentation strategies as well as threshold setting are important. In the future, this benchmark database could also be used to evaluate multi-task joint learning methods. we hope the provided benchmark database as well as the challenge setup could contribute to the development of personalized wake-up word detection techniques.

7. Acknowledgements

This research is funded in part by the National Natural Science Foundation of China (61773413), the Fundamental Research Funds for the Central Universities (2042021kf0039), Key Research and Development Program of Jiangsu Province (BE2019054), Science and Technology Program of Guangzhou City (201903010040,202007030011) and Six Talent Peaks Project in Jiangsu Province (JY-074)

[6] http://openslr.org/resources.php

8. References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*, 2014, pp. 4087–4091.

[2] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015.

[3] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Artificial Neural Networks*, 2007, pp. 220–229.

[4] M. Wöllmer, B. Schuller, and G. Rigoll, "Keyword spotting exploiting long short-term memory," in *Speech Communication*, 2013, pp. 252–265.

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

[6] H. Wu, Y. Jia, Y. Nie, and M. Li, "Domain aware training for far-field small-footprint keyword spotting," *Proc. Interspeech*, pp. 2562–2566, 2020.

[7] B. Liu, S. Nie, Y. Zhang, S. Liang, Z. Yang, and W. Liu, "Focal loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *Proc. ICASSP*, 2019, pp. 6361–6365.

[8] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Proc. ICASSP*, 2015, pp. 4704–4708.

[9] M. Sun, D. Snyder, Y. Gao, V. K. Nagaraja, M. Rodehorst, and S. Panchapagesan, "Compressed time delay neural network for small-footprint keyword spotting." in *Proc. Interspeech*, 2017, pp. 3607–3611.

[10] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Proc. Interspeech*, 2018, pp. 3743–3747.

[11] J. Hou, L. Zhang, Y. Fu, Q. Wang, Z. Yang, Q. Shao, and L. Xie, "The npu system for the 2020 personalized voice trigger challenge," *arXiv preprint arXiv:2102.13552*, 2021.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, 2018, pp. 4510–4520.

[13] "Res-se-net: Boosting performance of resnets by enhancing bridge-connections," *arXiv preprint arXiv:1902.06066*, 2019.

[14] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020, pp. 2977–2981.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[17] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[18] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. CVPR*, 2018, pp. 5265–5274.

[19] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation." in *Proc. Interspeech*, 2019, pp. 4045–4049.

[20] L. Primewords Information Technology Co., "Primewords chinese corpus set 1," 2018, https://www.primewords.cn.

[21] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *Proc. ICASSP*, 2020, pp. 7604–7608.

[22] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *Proc. ICASSP*, 2020, pp. 7609–7613.

[23] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *Proc. TASLP*, pp. 1038–1051, 2020.

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

[25] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5220–5224.

[26] C. S. Greenberg, V. M. Stanford, A. F. Martin, M. Yadagiri, G. R. Doddington, J. J. Godfrey, and J. Hernandez-Cordero, "The 2012 nist speaker recognition evaluation." in *Proc. Interspeech*, 2013, pp. 1971–1975.