

# A Two-Stage Query-by-example Spoken Term Detection System for Personalized Keyword Spotting

Yechen Wang\* Yan Jia\* Murong Ma\* and Ming Li \*

\* Duke Kunshan University, Kunshan, China

E-mail: ming.li369@dukekunshan.edu.cn

**Abstract**—In this paper, we propose a two-stage personalized keyword spotting system. Our implementation consists of a two-stage keyword spotting system based on query-by-example spoken term detection and speaker verification. We employ two different detection algorithms in our proposed keyword spotting system. The first stage adopts subsequence dynamic time warping for template matching based on frame-level language-independent bottleneck feature and phoneme posterior probability. We use a sliding window template matching algorithm based on acoustic word embeddings to further verify the detection from the first stage. As a result, our KWS system achieves an average score of 0.61 on the feedback dataset, which outperforms the baseline system by 0.25.

## I. INTRODUCTION

Keyword spotting (KWS), in terms of speech level, is a task that detects whether a predefined word or phrase has appeared in continuous speech. It is commonly used as the primary technique for low resource trigger systems and speech-based document analysis. More recently, KWS has been widely applied to our daily life, such as the wake-up word detection module for speech assistants on mobile phones, vehicles, and smart speakers. Those speech assistants are triggered by predefined keywords, like "Hey, Cortana," "Alexa," and "Hey, Siri," spoken by the owner. Such applications raise the need for a customized KWS system that could detect the keyword and identify the target speaker's voice simultaneously. To this end, more attention has been paid to develop a KWS system that responds to a particular speaker in recent research [1].

Conventionally, the KWS system consists of a large vocabulary continuous speech recognition (LVCSR) module, followed by a keyword spotting module that searches for keywords in the lattice generated by the LVCSR module [2]. The LVCSR module uses a large amount of audio-text pairs to train a traditional automatic speech recognition model that generates lattice, which contains the decoding information of the given speech. The following KWS module makes an index for the lattice in the lattice and searches keywords accordingly. This method provides a high accuracy approach while allowing us to customize the keywords without retraining the model. However, the lattice generated by the LVCSR module could be very complex with redundant information, which might result in inefficiency in real applications. Also, this approach has relatively low performance when it comes to out-of-vocabulary

(OOV) words because of its high dependency on the LVCSR module.

Another widely used approach for KWS is the query-by-example spoken term detection (QbE-STD) system. The QbE-STD method detects keywords through efficient template matching based on linguistic features extracted from the speech. Typically, the QbE-STD method contains two steps, known as feature extraction and template matching. For the first step, we extract the feature representing the content of the reference keyword audio segment. Various features have been investigated as the representation in the literature. For example, supervised features like language-independent bottleneck feature (BNF) and phoneme posterior probability (PPP) is adopted and yielded relatively good performance [3]. So as unsupervised features such as Mel-frequency cepstrum coefficients (MFCC) [4] and acoustic word embedding (AWE) [5], [6], [7], [8], [9]. The keyword detection is completed by the second step using template matching algorithms. Given a pair of features extracted respectively from the template speech and the evaluated speech, matching algorithms based on dynamic time warping (DTW) [10], such as segmental dynamic time warping [11] and subsequence dynamic time warping (SDTW) [11], [12], [13], are applied to measure the content similarity of the pair. The detection result is made according to the similarity score obtained from the matching algorithms. When handling the multilingual, multi accent, and various keywords situation, the QbE-STD is efficient both in time and accuracy. In this case, we choose the QbE-STD approach to develop our entry for the personalized keyword spotting system.

Unlike the baseline QbE-STD system proposed in [14] which only uses a one-stage QbE-STD system, we proposed a two-stage QbE-STD approach as our KWS system. Each stage contains a different QbE-STD system that consists of a feature extraction module and a template matching module. The first stage uses a BNF+PPP feature extractor in frame-level and an SDTW template matching algorithm. The second stage uses a sequence-level AWE feature extractor with a sliding window template matching algorithm. The search content that passes through both two stages will be considered the appearance of the keyword. To achieve personalized keyword spotting, we employed the SV system proposed in [1] to determine whether the given speech and the template come from the same speaker. Our system obtains a final score of 0.61, with an average miss

rate (MR) of 0.29 and an average false alarm rate (FAR) of 0.036 on the feedback dataset.

The paper is organized as follows. The task definition and proposed dataset are presented in section 2. Section 3 describes the detailed implementation of our system, and our experiment is described in section 4. Finally, the conclusion is provided in section 5.

## II. TASK DEFINITION AND PROPOSED DATASET

### A. Task Definition

All the task settings completely follow the Auto-KWS 2021 challenge description in [14]. We are focusing on building a personalized keyword spotting system that will be triggered if and only if the predefined keyword spoken by the target speaker is detected. This task consists of multilingual, multi-accent, and various keywords scenarios. The computational resources are also limited. The evaluation process uses a four-core CPU with 26G memory, 100G disk, and an NVIDIA Tesla P100 GPU. The computational budget during the evaluation process is 30 mins of initialization, 5 mins of enrollment, and 1 min + 0.25 \* total test duration. More details about the task definition and baseline implementation could be found in [14].

### B. Proposed Dataset

The training dataset provided by the Auto-KWS challenge organizer contains speech from 100 speakers recorded by mobile phones at a near-field around 0.2 meters. The audio has a single-channel 16-bit stream, and the sample rate is 16kHz. For each speaker, there are 10 enrollment utterances which contain the keyword, and a few others utterance that does not contain the keyword. Data augmentation is applied during the experiment to obtain more training data and improve the model accuracy and robustness. The data augmentation methods include perturbing the speed and the volume of the speech, adding noise, and splicing processing for the short speech audios during enrollment. The practice dataset, which contains speech audio from 5 speakers, is used in the evaluation. We will also show the evaluation result on the feedback dataset, which contains 20 speakers given by the automatic evaluation system provided by the Auto-KWS challenge organizer. There is no restriction on using other datasets in this task. Therefore we also include multiple Chinese corpora on OpenSLR. More details could be found in the experiments section.

## III. SYSTEM DESCRIPTION

### A. KWS System

We adopted a two-stage QbE-STD structure for our KWS system. In each stage, we applied a separated QbE-STD system with different feature extractors and template matching algorithms. When the first-stage model detects the keyword, the speech segment containing the keyword is fed to the second-stage model for another check. The second stage uses a sequence-level model with higher accuracy, but the output keyword time stamps of the second stage model are not as accurate as those of the first stage. Hence when the second stage also confirms the detection of the keyword, the keyword

segment detected by the first-stage model is sent to the speaker verification module as its timing information is more accurate.

1) *First-stage Model*: In general, feature extraction and template matching are used in the QbE-STD based KWS system. In the first stage, we use an acoustic model to extract the BNF and PPP feature. The time delay neural network (TDNN) based acoustic model has usually been applied in automatic speech recognition (ASR) tasks and achieves state-of-the-art performance. Therefore, we employ an acoustic modeling method based on the TDNN trained with frame-level training criteria.

2) *SDTW Template Matching Module*: In the template matching step, a DTW-based multiple templates strategy is used in the first-stage KWS system[15]. Since the QbE-STD tasks usually interfere with extraneous factors like channel variance, the templates fusion method has been widely used in the QbE-STD based system. Firstly, one of the prepared templates is chosen randomly as the master template, and then we apply the DTW algorithm to align the rest templates and get the shortest path. Finally, we compute the average of these aligned points in the shortest path and get the example template of the keyword. This fusion method allows us to obtain a more representative template though combining all the templates.

Traditional DTW [11] requires the start and the end time point of two sequences must be strictly aligned. In this task, we employ SDTW based algorithm [12], [13]. It can find a subsequence that does not necessarily go through the end time point, which optimally fits the spoken query in the search content. The Euclidean distance is used in this SDTW template matching module. The alignment result is used as a keyword time stamp in future steps as the features we extracted in frame-level usually provide higher accuracy in the time axis during template matching.

3) *Second-stage Model*: The second-stage keyword detection is activated after a successful trigger by the first stage. Similarly, we use two modules: an AWE system for feature extraction and a sliding window template matching method. Our AWE system uses a similar structure in [9]. It is trained with sequence-level training criteria. The network consists of a combination of a convolutional neural network (CNN), a global average pooling layer (GAP), and a fully connected layer in order. The log filter-bank energies (Fbank) of individual words are extracted and fed into the network. The CNN structure is based on a residual neural network (ResNet) as it has been proved efficient in structuring deep neural networks. A global average pooling (GAP) layer is applied as an aggregator over the three-dimensional output sequence. It computes the global mean feature values over the time and frequency axes. The output of the GAP layer then goes through the fully-connected (FC) layer. We use the cross-entropy loss to optimize the system. The block softmax layer is also introduced in each ResNet block to better handle the multilingual scenario [3], [16]. After the system is well-trained, we obtain the acoustic embedding feature from the output of the GAP layer.

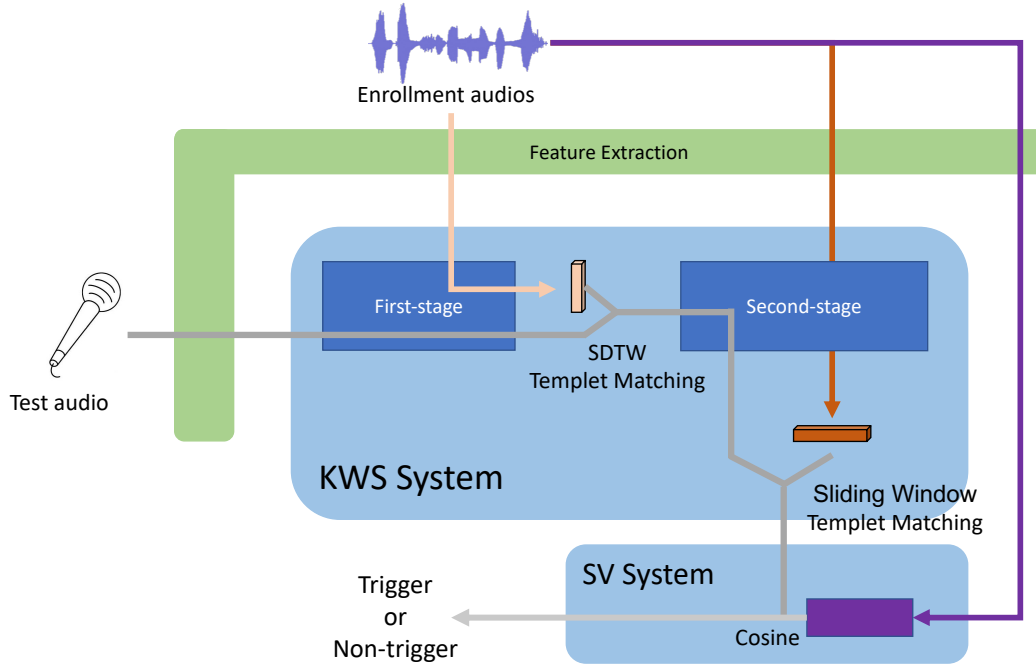


Fig. 1. The overview framework of the proposed system.

## B. Speaker Verification System

1) *Sliding Window Template Matching Module*: We choose a sliding window method and cosine distance as our template matching scheme. First, we pad or clip the keyword audio to 0.8 seconds, and use the same value as the fixed window size to convert the search content into a segment sequence  $y_1, y_2, \dots, y_t$ . Then each segment is fed into the AWE feature extractor we trained to generate a sequence of acoustic word embedding features  $f(y) = (f(y_1), f(y_2), \dots, f(y_t))$ . For each input, the cost is calculated between a segment sequence of the input  $f(x)$  and template  $f(y)$  following the equation:

$$Cost(x, y) = \min(1 - \frac{f(x) \cdot f(y_i)}{\|f(x)\|_2 \|f(y_i)\|_2}), i = 1, 2, \dots, T$$

It generates a score sequence with respect to time. The influence of random noise is removed by applying a simple moving average which could smooth the sequence by dividing the sum of a fixed number of continuous scores by the number of frames for the time involves. In addition, a template fusion method is also used in this module in order to find a more representative template. Since the feature extracted by the AWE system has a fixed length in this stage, we take the average of the templates as the fusion template.

We use a similar model structure as it is proposed in [17]. Specifically, our speaker verification system consists of a front-end feature extractor, a statistic pooling layer, and a back-end classifier. ResNet34 [18] with SE-block [19] is used as the feature extractor. An attentive statistics pooling (ASP) [20] is used as the encoding layer, which has been proved efficient in

detecting long-term speaker feature variations. For the back-end classifier, we use the AM-Softmax [21].

## C. Speaker Dependent KWS system

Our proposed system consists of a two-stage KWS system and a speaker verification system described above. To achieve personalized KWS, as shown in Figure 1, we design a speaker-dependent KWS system that only response to the target speaker when it detects the keyword. We extract three utterances' embedding by the target speaker during the enrollment stage and save them as the embedding vector template for the enrollment speaker.

The general procedure of the system is as follows: First, in the two-stage KWS system, we save the timing information generate by the first stage frame-level SDTW alignment result. We use this information to cut a fixed-length segment of the speech features vector and fed it into the speaker verification system. The SV system then uses this vector to obtain the speaker embedding by feeding this vector into the SV model we described above. The cosine similarity between this speaker embedding and the template we saved during the enrollment phase is compared with a threshold to determine if the input speech comes from the enrolled speaker. Together with the result of the two-stage KWS system and the result of the SV system, we consider an input speech to be positive if and only if it can successfully pass both the KWS system and the SV system. The general system diagram is shown in Figure 1.

#### IV. EXPERIMENTS

We separated the training process of the two-stage KWS system and the SV system. Two systems are optimized separately.

##### A. Implementation Details

1) *Keyword Spotting*: For our two-stage KWS system, in the first stage, we trained a frame-level TDNN based acoustic model on 40-dim MFCC features with a 25ms window length and a shift of 10ms. The training dataset includes multiple Chinese corpora on OpenSLR including Aidatatang [22], Aishell [23], MagicData [24], Primewords [25], ST-CMDS [26] and THCHS-30 [27]. The datasets we have used for training the acoustic model are shown in Table I. We trained a 3-gram language model using all the training transcriptions we have in the dataset. The lexicon is the CC-CEDIT Chinese dictionary expanded by Grapheme-to-Phoneme (G2P). The training starts by using a small part of data to accelerate the training procedure of the GMM model and then employ speaker adaptive training using all the datasets listed above. Finally, a Chain model is trained and evaluated while the PPP and BNF features are extracted from the final Chain model in the way of online decoding using Kaldi scripts [28]. We stack the BNF and PPP features together for the SDTW algorithm computation. We use a threshold here to decide if the input speech contains the keyword. We also save the timestamps on the shortest path for the later SV system as its timing information is more accurate.

TABLE I  
THE DATA USED FOR TRAINING THE ACOUSTIC MODEL

Dataset	Total hours
Aidatatang	140
Aishell	151
MagicData	712
Primewords	99
ST-CMDS	110
THCHS-30	26

In the second stage, we use the 64-dimensional Fbank energies as the input acoustic feature for the AWE model. We use 0.8 seconds as the window length and extract acoustic features on this window. The proposed neural network architecture is shown in Table III as L stands for input size.

We train the whole neural network for 80 epochs with categorical cross-entropy loss optimized by Stochastic Gradient Descent (SGD) with Nesterov momentum equal to 0.9. We initially set the learning rate equal to 0.1 and gradually decrease its value every time the loss stops decreasing. The procedure of the sliding window template matching has been introduced above. The second threshold is used in the second stage to decide if the input speech contains the keyword.

2) *Speaker Verification*: The general experimental of our SV system has the same procedure as in [1]. We pre-train our model by using data from SLR38 [26], SLR47 [25], SLR62 [22], SLR82 [29], SLR85 [30] on OpenSLR. The datasets we have used for pre-training the SV system are shown in Table

IV. We also add MUSAN [31] and RIRs-NOISES [32] as noise in the training set to make our model more robust. We set the signal-to-noise ratio (SNR) between 0 to 20 dB while pre-training and 0 to 15 dB while fine-tuning. The method in [17] is also applied to add reverberation to the data. We trained our model for 50 epochs during the pre-training process with an SGD optimizer together with a batch size of 256 and set the initial learning rate equal to 0.01 and decreases 0.1 after every 20 epochs. We fine-tuned our model for 20 epochs with a learning rate of 0.001. The third threshold is used here to decide if the input speech comes from the target speaker.

##### B. Evaluation

###### 1) Evaluation Metrics and Determination of Thresholds:

The metrics defined by the Auto-KWS organizers calculate the final score from a weighted sum of the MR and the FAR by using the equation below:

$$score_i = MR + \alpha \times FAR$$

where  $\alpha$  is a factor used to adjust the cost of MR and FAR. The lower the  $score_i$ , the better we consider the model is. Since we can adjust our three thresholds to make the trade-off between the MR and FAR, we fine-tune our model to achieve lower  $score_i$  and higher performance. We designed a development dataset in order to fine-tune our three thresholds. For the KWS system, for each target speaker in the Auto-KWS challenge training set, we directly select 5 enrollment utterances that contain the keyword and 20 other utterances that do not contain the keyword and randomly splice them together to create a new development dataset. For the SV system, we generate around 2800 trials using the Auto-KWS challenge training set to determine the threshold according to EER (Equal Error Rate) and minDCF [33] performance. We finally choose  $\alpha = 9$ , which is determined under the assumption that the positive samples account for 10 percent of all samples. We determine the thresholds of our system by using the development set to find the optimal thresholds which could minimize our  $score_i$ .

2) *SV Result*: The performance of our SV system on development data is shown in Table V. We use the development set to determine the threshold of the speaker verification system. The mean threshold of EER and minDCF[33] denoted as  $(threshold_{EER} + threshold_{minDCF})/2$  have been used in our system.

3) *Results of the Overall System*: Our proposed system on the feedback dataset achieves an average score of 0.611. The detailed result is shown in Table II. From the table, we can obtain the following observations from our system. First, using a more complex structure in the KWS system can achieve better results than the baseline systems. Our system achieves better performance than the baseline systems because we adopt the complex system structure of two-stage KWS models and large-scale speaker verification models. Second, only using the original training data to train our model makes it hard to generalize the model to the development and evaluation sets, resulting in a very low recall. Thus, the method to

TABLE II  
DETAILED RESULT ON PRACTICE AND FEEDBACK DATASET

Model	Dataset	Average Score	Average MR	Average FAR
Our System	Practice Dataset	0.240	0.240	0
	Feedback Dataset	0.611	0.289	0.0359
Baseline System 1	Practice Dataset	-	-	-
	Feedback Dataset	0.859	0.443	0.046
Baseline System 2	Practice Dataset	-	-	-
	Feedback Dataset	1.695	0.481	0.135

TABLE III  
THE ARCHITECTURE FOR AWE SYSTEM

Layer	Output Size	Downsample	Channels	Blocks
Conv1	$16 \times \frac{L}{4}$	False	64	-
Res1	$16 \times \frac{L}{4}$	False	64	3
Res2	$8 \times \frac{L}{8}$	True	128	4
Res3	$4 \times \frac{L}{16}$	True	256	6
Res3	$2 \times \frac{L}{32}$	True	512	3
GAP	512	-	-	-
Output	Number of Words	-	-	-

TABLE IV  
THE DATA USED FOR PRE-TRAINING

	Number of Speakers	Total hours	Utterances
SLR38	855	100+	102600
SLR47	296	100+	50384
SLR62	600	200	237265
SLR62	274	1000	130108
SLR85	340	1500	108678

determine the threshold is an important factor affecting the final score. The ad-hoc average threshold of EER and minDCF can improve system performance.

## V. CONCLUSIONS

In this paper, we introduced a personalized keyword spotting system. Our system consists of a two-stage KWS system and an SV system. Although the two systems are optimized separately, we managed to find an efficient way to work together and achieve personalized KWS. For the two-stage KWS system, we employed a BNF and PPP feature extractor and an SDTW template matching method as the first stage and an AWE feature extractor with a sliding window template matching method as the second stage. Different fusion methods are applied to produce templates in different stages to improve performance. We also introduced data augmentation to improve the accuracy and robustness of the system and designed the development dataset to optimize our score. Under the restriction of computational resources, we successfully outperform the baseline system by 0.25 on the average score, and our system reaches the average score of 0.61 on the feedback dataset.

## VI. ACKNOWLEDGEMENTS

This research is funded in part by the National Natural Science Foundation of China (61773413), the Fundamental Research Funds for the Central Universities

TABLE V  
PERFORMANCES OF SPEAKER VERIFICATION MODEL ON THE DEV SETS (EER[%] AND MINDCF)

Model	EER	minDCF
SV System	4.85	0.39

(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).

## REFERENCES

- [1] Y. Jia, X. Wang, X. Qin, Y. Zhang, X. Wang, J. Wang, and M. Li, "The 2020 Personalized Voice Trigger Challenge: Open Database, Evaluation Metrics and the Baseline Systems," *arXiv preprint arXiv:2101.01935*, 2021.
- [2] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in lowresource languages," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 8560–8564.
- [3] J. Hou, C.-C. L. Van Tung Pham, C.-C. Leung, L. Wang, H. Xu, H. Lv, L. Xie, Z. Fu, C. Ni, X. Xiao et al. "The NNI Query-by-Example System for MediaEval 2015," *MediaEval*, 2015.
- [4] P. Yang, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection," *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [5] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 4950–4954.
- [6] S. Settle and K. Livescu, "Discriminative acoustic word embeddings: Tecurrent neural network-based approaches," *2016 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2016, pp. 503–510.
- [7] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," *arXiv preprint arXiv:1706.03818*, 2017.
- [8] K. Levin, A. Jansen, and B. Van Durme, "Segmental acoustic indexing for zero resource keyword search," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 5828–5832.
- [9] M. Ma, H. Wu, X. Wang, L. Yang, J. Wang, and M. Li, "Acoustic Word Embedding System for Code-Switching Query-by-example Spoken Term Detection," *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.
- [10] V. Velichko and N. Zagoruyko, "Automatic recognition of 200 words," *International Journal of Man-Machine Studies*, vol. 2, no. 3, 1970, pp. 223–234.
- [11] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, 2007, pp. 69–84.
- [12] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for query-by-example spoken term detection," *2013 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2013, pp. 1–6.
- [13] H. Wu, M. Li, Z. Cai, and H. Zhong, "Unsupervised query by example spoken term detection using features concatenated with Self-Organizing Map distances," *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, IEEE, 2018, pp. 1–5.

- [14] J. Wang, Y. He, C. Zhao, Q. Shao, W.-W. Tu, T. Ko, H.-y. Lee, and L. Xie, "Auto-KWS 2021 Challenge: Task, Datasets, and Baselines," *arXiv preprint arXiv:2104.00513*, 2021.
- [15] G. Chen, C. Parada, and T. N. Sain, "Query-by-example keyword spotting using long short-term memory networks," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 5236–5240.
- [16] K. Veselý, M. Karafiát, F. Grézl, František, M. Janda, and E. Egorova, "The language-independent bottleneck features," *2012 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2012, pp. 336–341.
- [17] 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," *arXiv preprint arXiv:2003.11982*, 2020.
- [18] K. He, X. Zhang, S. Ren, and J. Su, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pg. 7132–7141.
- [20] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.
- [21] 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," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.
- [22] "aidatatang\_200zh, a free Chinese Mandarin speech corpus by Beijing DataTang Technology Co., Ltd ( [www.datatang.com](http://www.datatang.com) )," <https://www.openslr.org/38/> Accessed July 21, 2021.
- [23] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, IEEE, 2017, pp. 1–5.
- [24] Magic Data Technology Co., Ltd., 2019, <http://www.openslr.org/68/> Accessed July 21, 2021.
- [25] Primewords Information Technology Co., Ltd., "Primewords Chinese Corpus Set 1," 2018, <https://www.primewords.cn> Accessed July 21, 2021.
- [26] ST-CMDS-20170001\_1, Free ST Chinese Mandarin Corpus, <https://www.openslr.org/38/> Accessed July 21, 2021.
- [27] D. Wang and X. Zhang, "Thchs-30: A free chinese speech corpus," *arXiv preprint arXiv:1512.01882*, 2015.
- [28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society, 2011.
- [29] 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," *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7604–7608.
- [30] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7609–7613.
- [31] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484v1*, 2015.
- [32] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 5220–5224.
- [33] 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," *INTERSPEECH*, 2013, pp. 1971–1975.