Acoustic Word Embedding System for Code-Switching Query-by-example Spoken Term Detection

Murong Ma[1]*, Haiwei Wu*[1]*, Xuyang Wang*[3]*, Lin Yang*[3]*, Junjie Wang*[3] *and Ming Li*[1,2]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]School of Computer Science, Wuhan University, China
[3]AI Lab of Lenovo Research, Beijing, China

ming.li369@dukekunshan.edu.cn

Abstract

In this paper, we propose a deep convolutional neural network-based acoustic word embedding system for code-switching query by example spoken term detection. Different from previous configurations, we combine audio data in two languages for training instead of only using one single language. We transform the acoustic features of keyword templates and searching content segments obtained in a sliding manner to fixed-dimensional vectors and calculate the distances between them. An auxiliary variability-invariant loss is also applied to training data within the same word but different speakers. This strategy is used to prevent the extractor from encoding undesired speaker- or accent-related information into the acoustic word embeddings. Experimental results show that our proposed system produces promising searching results in the code-switching test scenario. With the employment of variability-invariant loss, the searching performance is further enhanced.

Index Terms: convolutional neural network, acoustic word embedding, code-switching, query by example

1. Introduction

Spoken term detection (STD) [1, 2] is a technique to detect specific words in streaming audio or audio files. With the development of Internet media and smart devices, the demand for searching keywords in audio signal and voice control increased rapidly.

Query by example (QbE) is a particular case of the STD problem, whose task is to find out the occurrences of a keyword given its audio samples. A typical solution for this task is applying Dynamic Time Warping (DTW) [3] or its variants on frame-level features extracted from keyword templates and searching content [4]. Both supervised [5, 6] and unsupervised [7, 8] methods are explored to extract frame-level features by many researchers. Unsupervised features contain traditional acoustic features like filter-bank energy (Fbank) and Mel-Frequency Cepstrum Coefficients (MFCC) [9], as well as features obtained from unsupervised models like GMM by computing the posterior probabilities of the components [8]. Supervised frame-level features include phonetic features extracted by a neural network like language-independent Bottleneck feature (BNF) and phone posterior probabilities [4, 10]. DTW and its variants, such as segmental DTW [3] and subsequence DTW [3, 11, 12], are then employed to find out the most matching feature sequences in the searching content and keyword audio.

In recent years, there has been increased interest in applying acoustic word embedding (AWE) methods to QbE-STD tasks [13, 14, 15, 16]. Acoustic word embeddings are segment-level features extracted from the penultimate or final layer of the word-discriminative neural network. The network projects the features of audio segments to a fixed-dimensional vector space. Researchers have explored different network structures in this task, including convolutional neural networks (CNNs) [13] and recurrent neural networks (RNNs) [14, 15], which show superior performance to traditional methods. After training, embeddings of the same keyword have smaller distances with each other, and embeddings of the different keywords have more considerable distances. Then, a sliding analysis window [17] is taken to detect the occurrences of keywords.

Most of the AWE systems focus on the single language scenario, which means that only one language is spoken in the audio. However, speakers may switch between several languages in real-life situations. Code-switching is a practice of alternating between two or more languages in the context of a single conversation. It is a common phenomenon in many areas of the world, especially in second language education. However, to our knowledge, most studies focusing on the code-switching scenario are automatic speech recognition (ASR) tasks oriented and fewer studies on the QbE-STD task. This motivates us to explore the QbE-STD task from the code-switching testing data point of view.

In this paper, we propose a multi-language deep acoustic word embedding system with multiple templates. Besides the word discrimination loss, an auxiliary variability-invariant loss is also proposed to make the system generalize better on searching content and keywords spoken by different speakers. The word discrimination loss learns to encode word embeddings with labeled word audio, and the variability-variant loss aims to further decrease the distance between embeddings of the same keyword spoken by different speakers with or without accents. A similar method has also been used in fields such as speaker recognition [18], speech recognition [19] and far-field speaker recognition [20]. Our method selects audio data in English and Chinese for our training instead of using training data from a single language in other configurations. With data in two languages, we train a deep convolutional neural network for audio word discrimination. The trained model is used to extract embeddings for both keyword audio and searching content, and a sliding window accompanied by cosine distance computation is applied to detect the keywords. We also utilize the averaged template to reduce the within keyword variabilities. Recently, this idea has also been applied in the QbE-STD system [21].

2. Baseline systems

Generally speaking, there are two main steps in the traditional query by example (QbE) system, feature extraction, and template matching. In our baseline system, we utilize the DNN

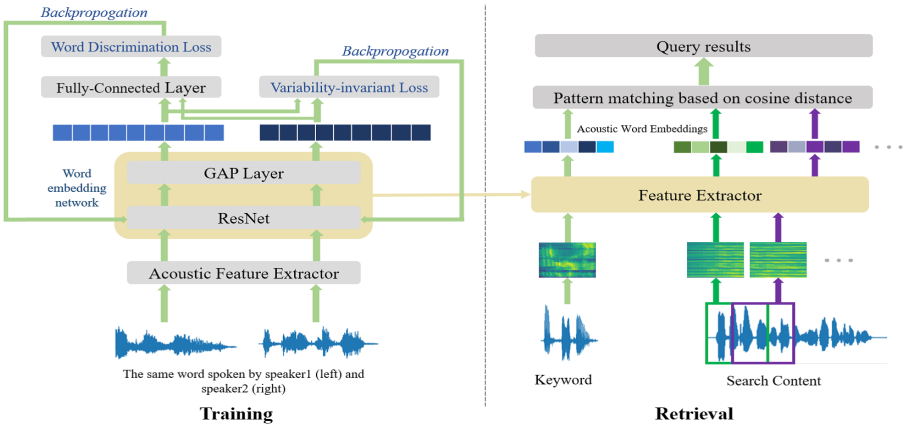

Figure 1: *The pipline of the whole AWE system for the QbE-STD task*

based phone posterior probabilities (PPP) as feature together with subsequence dynamic time warping (subsequence-DTW) algorithm for template matching.

2.1. DNN phone posterior probabilities

DNN based acoustic modeling is usually applied in automatic speech recognition (ASR) task and achieves state-of-the-art performance. In our study, we employ an acoustic modeling method based on the time delay neural network (TDNN) to obtain the phone posterior probabilities (PPP) as the baseline system. To make the baseline system suitable in a code-switching scenario, we separately train a Chinese acoustic model and an English acoustic model for feature extraction. The final PPP feature sequences can be fetched from the output layer of the trained TDNN acoustic model.

2.2. Template matching

The matching algorithm and fusion method used in the baseline system will be introduced in this section.

a) Subsequence-DTW. Traditional DTW [3] requires the start and the end of two sequences must be strictly aligned. Instead, in our cases, we employ subsequence-DTW (S-DTW) [11, 12] which allows us to find a subsequence within the search content that most optimally fits the spoken query.

b) Fusion method. In this work, we employ multiple templates strategy in baseline systems. Fusion on templates is widely used to cope with the variability caused by extraneous factors like speaker variance in the QbE-STD task. In the baseline system, a DTW-based fusion scheme is applied to obtain a more representative example [22]. Specifically, in the first step, we randomly choose one from the prepared templates as the main template. Second, we apply the DTW algorithm to align each of the rest templates with the main template to get a warping path. Third, we calculate the average of the aligned points in the warping paths and obtain a more energetical representation of each keyword.

3. Acoustic Word Embedding

3.1. Network structure

Our network is a combination of a convolutional neural network structure, a global average pooling layer, and a fully-connected layer in sequence. The network is trained to classify different words and used as a feature extractor in the testing phase. We extract log filter-bank energies (Fbank) of individual words as the input acoustic feature. The CNN structure works as a local pattern extractor that maps the input feature sequences into a compressed high-level abstract tensor block with temporal order. We set up our deep CNN based on the popular residual neural network (ResNet) [23]. The corresponding parameters are described in Table 1.

By forwarding the feature sequences through the deep CNN structure, the acoustic features can be transformed into a three-dimensional feature map, which still has one dimension related to time. Then the global average pooling layer (GAP) acts as an aggregator over the entire sequence by computing the global mean feature values over the time and frequency axes. The output representation is then fed into the following fully-connected (FC) layer. Cross-entropy loss and auxiliary variability-invariant loss are employed to optimize the system, and the final acoustic embedding can be fetched from the output of the GAP layer. The whole procedure of AWE system for the QbE-STD task is depicted in Figure 1.

In addition to the traditional one softmax layer, we also employ block softmax layer, which has proved effective in multilingual BNF extraction [4, 10, 24]. The major difference of our study is that we employ block softmax on segment-level input targeted with words instead of frame-level input targeted with phonemes. The purpose of block softmax is to lead language-dependent information into the feature map by dividing the output layer into multiple blocks according to the language. Each block of the output layer corresponds to an individual language and is activated only if the input data is from the associated language. This mechanism can be implemented with an interval-based softmax function

Table 1: *ResNet structure. N/A: Not available*

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
Res4	$2 \times \frac{L}{32}$	True	512	3
GAP	512	N/A	N/A	N/A
Output	number of words	N/A	N/A	N/A

$$y_i = \frac{exp(a_i)}{\sum_{j=n_{l,b}}^{n_{l,e}} exp(a_j)}. \quad (1)$$

y_i denotes the posterior of the i-th output; a_i represents the i-th activation value and $n_{l,b}$ is the beginning index of the *l*-th language while $n_{l,e}$ is the ending index.

3.2. Variability-invariant Loss

Recently, variability-invariant loss has been employed in speaker recognition [18], speech recognition [19] and far-field speaker recognition [20]. For QbE-STD tasks, the person who speaks search content and keyword is usually different. Ideally, the embeddings of the same keyword spoken by different speakers should be identical to each other. However, the extractor usually encodes speaker-related information as a part of the word representation. To make it concentrate more on word discrimination, we use the variability-invariant loss for each word during the training phase. For each instance I_{w,p_1} with word label w and speaker label p_1 forwarded through the network, we randomly choose another instance I_{w,p_2} with the same word label but different speaker label in the training set. The word embeddings $e_{w,p_1}, e_{w,p_2} \in \mathbb{R}^d$ encoded by the extractor E are

$$\begin{aligned} e_{w,p_1} &= E(I_{w,p_1}) \\ e_{w,p_2} &= E(I_{w,p_2}), \end{aligned} \quad (2)$$

where d denotes the dimension of the word embeddings. The loss function is used to calculate the distance between e_{w,p_1} and e_{w,p_2}. In this paper, we investigate mean square error (MSE) regression loss as loss funtion

$$l_{MSE}(e_{w,p_1}, e_{w,p_2}) = \frac{1}{d}\|e_{w,p_1} - e_{w,p_2}\|_2^2, \quad (3)$$

which calculates the average square difference between two embeddings of the same word spoken by different speakers. $\|\cdot\|_2$ denotes the L_2 norm. The variability-invariant loss and the word discrimination loss, which is typically a cross-entropy loss, are jointly used to train the acoustic word embedding network and make the system more robust. The total loss function L_t can be represented as

$$\begin{aligned} L_t = l_{CE}(y_{w,p_1}, \hat{y}_{w,p_1}) + l_{CE}(y_{w,p_2}, \hat{y}_{w,p_2}) \\ + \alpha l_{MSE}(e_{w,p_1}, e_{w,p_2}). \end{aligned}$$

$\hat{y}$ denotes the logit output and y represents the ground truth value. The hyper-parameters α of the network are fixed according to the results of our pre-experiment.

3.3. Template matching

We employ cosine distance computation with a sliding window as our template matching scheme.

a) Sliding window. A fixed-size sliding window is applied to convert an utterance in search content into a segment sequence $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, ..., \boldsymbol{y}_T$. Each segment is fed into the trained deep CNN, and we can get a sequence of acoustic word embedding $f(\boldsymbol{y}) = (f(\boldsymbol{y}_1), f(\boldsymbol{y}_2), ..., f(\boldsymbol{y}_T))$ derived from the output of GAP layer. Also, we pad or clip the keyword audio to 0.8s, which is equal to the window size we applied. The input segment $\boldsymbol{x}$ is transformed into the embedding $f(\boldsymbol{x})$ with deep CNN. So far, we can compute the cost between a segment sequence of the search content $\boldsymbol{y}$ and a spoken query $\boldsymbol{x}$ as follows:

$$Cost(\boldsymbol{x}, \boldsymbol{y}) = \min\left(1 - \frac{f(\boldsymbol{x}) \cdot f(\boldsymbol{y}_i)}{||f(\boldsymbol{x})||_2 ||f(\boldsymbol{y}_i)||_2}\right), i = 1, ..., T. \quad (4)$$

b) Simple moving average. After the computation of cosine distance with the sliding window, a time-dependent score sequence is generated for each utterance to be searched. To reduce the influence of random fluctuation of the scores, we further employ simple moving average (SMA) to smooth the sequence. In SMA, output scores are calculated by taking the sum of recent scores and then dividing that by the number of involving frames for each point.

c) Multi-template. We also employ the multi-template strategy in our original QbE system for the same reason stated in the baseline system but with a different fusion strategy. Compared with the input features with arbitrary length in the baseline system, features extracted by the AWE system are fixed-dimensional, which allows us to fuse templates by simply taking average.

4. Experiments

4.1. Experiment setup

In the PPP + S-DTW baseline system, we trained three acoustic models with the following training set: a) Chinese data, b) English data as the first language (L1), and c) mixed English data of L1 and L2 (second language). The Chinese acoustic model (a) is trained with MFCC-pitch features from the AISHELL-2 Chinese dataset [25]. English acoustic model (b) is trained with MFCC features from Librispeech dataset [26]. English L2 and L1 mixed model (c) is trained with MFCC features from the Librispeech dataset (L1 for English) and MDT-ASR-A004 dataset [1] (L2 for English). The PPP features are extracted from the output of TDNN acoustic models implemented with Kaldi nnet3 scripts [27].

As for our proposed AWE system, we adopt 220k spoken word tokens for English and Chinese. Audio of 1459 English word types and 1956 Chinese word types are aligned from the Librispeech English (L1) dataset, MDT-ASR-A004 English (L2) dataset, and AISHELL-2 dataset. For evaluation, we select keyword audio templates from English (L1) dataset TIMIT [28], English (L2) dataset MDT-ASR-A004 and Chinese dataset THCHS30 [29]. 86 Chinese keywords and 52 English keywords, with five to ten templates for each word, are used. The word types chosen from L1 and L2 English datasets are the same. As the L2 English keywords for evaluation and training come from the same dataset, we specially split the dataset so that no audio spoken by one person appears both in training and testing set. And we utilize 1191 utterances of code-switching dataset from Datatang AI Dataset [2] where our chosen keywords appear as the testing data. In this dataset, the speaker may alternate language from Chinese to English in some words while speaking. Our task is to detect the occurrences of both Chinese and English keywords in these code-switching audio utterances.

We employ the 64-dimensional Fbank energies as input acoustic features for our AWE system. The neural network model is trained with categorical cross-entropy and variability-invariant loss as loss function and optimized by Stochastic Gradient Descent (SGD) with Nesterov momentum 0.9. The learning rate is first initialized as 0.1 and reduces when the loss stops decreasing. We train the model for 80 epochs, and after training, we extract embeddings for segments of keyword audio and search content from the penultimate layer of the network. The size of the sliding window is 0.8 seconds, which covers the length of most keywords.

[1] https://www.magicdatatech.com/goods/3309.html

[2] https://www.datatang.com

Table 2: *Performance of PPP + S-DTW, one and block softmax AWE systems without variability-invariant loss and one softmax with variability-invariant loss on code switching dataset*

System	KW lang and template types	Metrics		
		MAP	P@5	P@N
(a) CN PPP + S-DTW	CN	**0.795**	**0.820**	**0.464**
	EN (L1)	0.046	0.053	0.036
	EN (L2)	0.092	0.130	0.077
(b) EN (L1) PPP + S-DTW	CN	0.069	0.113	0.061
	EN (L1)	0.307	0.369	0.223
	EN (L2)	0.284	0.315	0.212
(c) EN (L1,L2) PPP + S-DTW	CN	0.144	0.206	0.113
	EN (L1)	0.418	0.407	0.266
	EN (L2)	0.747	0.726	0.460
(d) EN (L1,L2),CN PPP + S-DTW	CN	0.691	0.739	0.423
	EN (L1)	0.227	0.284	0.180
	EN (L2)	0.565	0.642	0.396
(e) Block Softmax AWE without V-I loss	CN	0.725	0.742	0.426
	EN (L1)	0.556	0.588	0.377
	EN (L2)	0.757	0.777	0.478
(f) One Softmax AWE without V-I loss	CN	0.701	0.746	0.422
	EN (L1)	0.570	0.596	0.384
	EN (L2)	0.769	0.596	0.490
(g) One Softmax AWE with V-I loss	CN	0.702	0.737	0.418
	EN (L1)	**0.634**	**0.665**	**0.414**
	EN (L2)	**0.804**	**0.838**	**0.534**

In our experiments, following previous researches [7, 17], we use MAP, P@5, P@N as our evaluation metrics. MAP (Mean Average Precision) refers to the mean of average precision for each keyword in search content. P@5 (Precision at 5) is the precision of the top 5 utterances retrieved by the system and P@N (Precision at N) is the precision of top N utterances, where N means the number of target keywords in search content.

4.2. PPP + S-DTW systems

As Table 2 shows, system (a) and system (b) are trained with Chinese and English (L1) datasets separately, and we can see that they produce better results on their own language while lower scores on the other language. And system (a) with Chinese templates achieves the best results among all methods, which means that the PPP + S-DTW system is suitable in a single language QbE-STD task.

From the results of (b), we can observe that the scores on both L1 and L2 English keywords are lower than expectations comparing with those Chinese words. The possible explanation is that the English audio words are spoken by Chinese speakers and have different kinds of accents. Besides, the audio context of the English words is still in Chinese, which may affect the searching performance of English words. To further investigate this phenomenon, in system (c), we train an English DNN model with both L1 English and L2 English audio data to reduce the influence of accent. The results of (c) are better than (b) on both L1 and L2 English keywords, which reveals that multi-condition training can reduce the gap of audio words caused by accent. Besides, the result on Chinese keywords achieved by system (c) is better than (b), which might be because L2 English (spoken by Chinese) words contribute to Chinese word representation.

To achieve better overall results on both Chinese and English words, in the system (d), we concatenate the PPP extracted with system (a) and (c) on the feature level. The result shows that the performance of concatenated features is more balanced than the PPP of a single language.

4.3. AWE systems with different softmax functions

From the view of system (e) and (f), we can see that our proposed systems produce competitive results over the baseline PPP + S-DTW systems on all types of keywords except result on Chinese keywords achieved by system (a). The results on L1 English words of the proposed AWE systems are much better than other systems, which proves that our proposed method has the potential to overcome the mismatch caused by accent between keyword templates and searching content to some extent. Also, system (e) and (f) get good performance on L2 English keywords. Our proposed system is suitable for the code-switching scenario, while we should also find that on Chinese words, the performance gap between the AWE system and PPP + S-DTW system (b) still exists. The PPP + S-DTW system is still robust on a single language searching task.

4.4. AWE systems with variability-invariant loss

In this work, the hyper-parameter of the variability-invariant loss is fixed at 0.8, according to our preliminary experimental results. Table 2 also shows the effectiveness of the usage of variability-invariant loss. We can see that system (g) has a considerably better result on English (both L1 and L2) keywords than Chinese keywords. The loss is employed on instances with the same word label but different speaker labels. In the English scenario, it minimizes the difference within a word between not only speakers but also accents.

5. Conclusions

In this paper, we propose an AWE QbE-STD system based on a deep convolutional neural network. We utilize training data of two languages to train deep neural networks with both one softmax and block softmax layer. Acoustic word embeddings are extracted from the penultimate layer of the network, and cosine distances are computed between embeddings of keyword audio segments and search content segments with sliding windows. Experimental results show that our proposed AWE system with one softmax or block softmax layer generates competitive results over the baseline PPP + S-DTW systems. Variability-invariant loss is employed to decrease the influence caused by speaker-related information, and the experiment result shows the effectiveness of this method.

6. Acknowledgements

This research is funded in part by the National Natural Science Foundation of China (61773413), Key Research and Development Program of Jiangsu Province (BE2019054), Six talent peaks project in Jiangsu Province (JY-074), Science and Technology Program of Guangzhou City (201903010040,202007030011) and Lenovo.

7. References

[1] Anupam Mandal, K. Prasanna Kumar, and Pabitra Mitra. Recent developments in spoken term detection: A survey. *International Journal of Speech Technology*, 17:183–198, 2014.

[2] Alex S. Park and James R. Glass. Unsupervised pattern discovery in speech. *Transactions on Audio Speech & Language Processing*, 16(1):186–197, 2008.

[3] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[4] Jingyong Hou, Van Tung Pham, Cheung-Chi Leung, Lei Wang, Haihua Xu, Hang Lv, Lei Xie, Zhong-Hua Fu, Chongjia Ni, Xiong Xiao, Hongjie Chen, Shaofei Zhang, Sining Sun, Yougen Yuan, Pengcheng Li, Tin Lay Nwe, Sunil Sivadas, Bin Ma, Eng Siong Chng, and Haizhou Li. The NNI query-by-example system for MediaEval 2015. In *MediaEval*, 2014.

[5] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez. High-performance query-by-example spoken term detection on the SWS 2013 evaluation. In *Proc. ICASSP*, pages 7819–7823, 2014.

[6] Cheung-Chi Leung, Haihua Xu, Jingyong Hou, Tung Pham, Hang Lv, Lei Xie, Xiong Xiao, Chongjia Ni, Bin Ma, Eng Chng, and Haizhou Li. Toward high-performance language-independent query-by-example spoken term detection for MediaEval 2015: Post-evaluation analysis. In *Proc. INTERSPEECH*, pages 3703–3707, 2016.

[7] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li. Unsupervised bottleneck features for low-resource query-by-example spoken term detection. In *Proc. INTERSPEECH*, pages 923–927, 2016.

[8] Y. Zhang and J. R. Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *ASRU*, pages 398–403, 2009.

[9] P. Yang, Cheung-Chi Leung, L. Xie, Bin Ma, and Haizhou Li. Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1722–1726, 2014.

[10] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 37(2):233–243, 1991.

[11] Xavier Anguera and Miquel Ferrarons. Memory efficient subsequence DTW for query-by-example spoken term detection. In *Proc. ICME*, pages 1–6, 2013.

[12] Haiwei Wu, Ming Li, Zexin Cai, and Haibin Zhong. Unsupervised query by example spoken term detection using features concatenated with self-organizing map distances. In *ISCSLP*, 2018.

[13] H. Kamper, W. Wang, and K. Livescu. Deep convolutional acoustic word embeddings using word-pair side information. In *Proc. ICASSP*, pages 4950–4954, 2016.

[14] Shane Settle and Karen Livescu. Discriminative acoustic word embeddings: Recurrent neural network-based approaches. *Spoken Language Technology Workshop (SLT)*, pages 503–510, 2016.

[15] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu. Query-by-example search with discriminative neural acoustic word embeddings. In *Proc. Interspeech*, pages 2874–2878, 2017.

[16] K. Levin, A. Jansen, and B. Van Durme. Segmental acoustic indexing for zero resource keyword search. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5828–5832, 2015.

[17] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Hongjie Chen, Bin Ma, and Haizhou Li. Learning acoustic word embeddings with temporal context for query-by-example speech search. In *Proc. Interspeech*, pages 97–101, 2018.

[18] D. Cai, W. Cai, and M. Li. Within-sample variability-invariant loss for robust speaker recognition under noisy environments. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6469–6473, 2020.

[19] D. Liang, Z. Huang, and Z. C. Lipton. Learning noise-invariant representations for robust speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 56–63, 2018.

[20] Jonathan Huang and Tobias Bocklet. Intel far-field speaker recognition system for voices challenge 2019. pages 2473–2477, 09 2019.

[21] Yougen Yuan, Zhiqiang Lv, Shen Huang, and Lei Xie. Verifying deep keyword spotting detection with acoustic word embeddings. 2019.

[22] G. Chen, C. Parada, and T. N. Sainath. Query-by-example keyword spotting using long short-term memory networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5236–5240, 2015.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[24] Karel Vesely, Martin Karafiat, Frantisek Grezl, Milos Janda, and Ekaterina Egorova. The language-independent bottleneck features. In *SLT Workshop*, 2013.

[25] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin ASR research into industrial scale. *arXiv preprint arXiv:1808.10583*, 2018.

[26] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *Proc. ICASSP*, pages 5206–5210, 2015.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[28] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

[29] Dong Wang and Xuewei Zhang. Thchs-30: A free chinese speech corpus. *arXiv preprint arXiv:1512.01882*, 2015.