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

Text Anchor Based Metric Learning for Small-footprint Keyword Spotting

Li Wang[1], *Rongzhi Gu*[1], *Nuo Chen*[1], *Yuexian Zou*[1,2,*]

[1] ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China

{1901213145, 1701111335, nuochen, zouyx}@pku.edu.cn

Abstract

Keyword Spotting (KWS) remains challenging to achieve the trade-off between small footprint and high accuracy. Recently proposed metric learning approaches improved the generalizability of models for the KWS task, and 1D-CNN based KWS models have achieved the state-of-the-arts (SOTA) in terms of model size. However, for metric learning, due to data limitations, the speech anchor is highly susceptible to the acoustic environment and speakers. Also, we note that the 1D-CNN models have limited capability to capture long-term temporal acoustic features. To address the above problems, we propose to utilize text anchors to improve the stability of anchors. Furthermore, a new type of model (LG-Net) is exquisitely designed to promote long-short term acoustic feature modeling based on 1D-CNN and self-attention. Experiments are conducted on Google Speech Commands Dataset version 1 (GSCDv1) and 2 (GSCDv2). The results demonstrate that the proposed text anchor based metric learning method shows consistent improvements over speech anchor on representative CNN-based models. Moreover, our LG-Net model achieves SOTA accuracy of 97.67% and 96.79% on two datasets, respectively. It is encouraged to see that our lighter LG-Net with only 74k parameters obtains 96.82% KWS accuracy on the GSCDv1 and 95.77% KWS accuracy on the GSCDv2.

Index Terms: keyword spotting, long-term information, metric learning, small-footprint, text anchor

1. Introduction

With speech technology development, speech assistants can help people solve affairs more efficiently, such as querying weather and controlling air conditioning. People increasingly enjoy the convenience of the hands-free experience. Keyword Spotting (KWS) is the beginning of the human-computer speech interaction, aims at distinguishing between each of the target keywords of interest and non-target sounds such as general speech (non-target words) and noises [1].

Recently, researchers have improved the generalization performance of models in terms of data enhancement [2], loss function [3, 4], automatic gain control [5], negative sample mining [6], and metric learning [1, 7]. In particular, the metric learning based on triplet loss [1, 7] has demonstrated excellent performance. The KWS models are trained using triplet consists of an $anchor$ sample, a $positive$ sample from the same class with the $anchor$, and a $negative$ sample from a different class. The objective of the network training is to minimize the distance between the embeddings of the $anchor$ and the $positive$ sample while maximizing the distance between the embeddings of the $anchor$ and the $negative$ sample. It can be seen that the performance is much dependent on the choice of the anchor. However, due to data limitations, the speech anchor is highly susceptible to the acoustic environment and speakers, making the anchor embedding more variant during training. In addition, deviated speech anchors may make the KWS models suffering from a local optimality issue, which leads to the degradation of the performance.

On the other hand, Deep neural networks (DNNs) have recently proven to yield efficient small-footprint solutions for KWS [8, 9, 10, 11, 12, 13, 14, 15, 16]. In particular, more advanced architectures, such as Convolutional Neural Networks (CNNs), have been applied to solve KWS problems under limited memory footprint as well as computational resource scenarios, showing excellent accuracy. Most CNN-based KWS models receive features, such as Mel-Frequency Cepstral Coefficient (MFCC), as a 2D input. However, such 2D CNN-based KWS models struggle with capturing the dependency between low and high frequencies with the relatively shallow network. To address this problem, [13, 15] utilize temporal convolution to capture low and high-frequency features in a shallow network, which achieves the best KWS performance. Despite their success, due to local region perception and weight sharing characteristics of CNNs, the long-term temporal information may not be considered. Figure 1a shows the feature maps produced by a representative 1D-CNN-based KWS model, TC-ResNet14-1.5 [13]. It is noted that the saliency regions are distributed sparsely, which implies that 1D-CNN focuses on the short-term temporal information. In fact, it is crucial to capture the long-term temporal information for KWS, considering that the characteristics of keywords are usually different on the time scale.

In this study, we aim at boosting the performance of the small-footprint KWS model. To improve the stability of the anchor, we replace the speech anchor with the text anchor and propose a text anchor based metric learning method for KWS. Specifically, target keywords and non-target words are encoded in the text anchors by pretraining models (like BERT [17]). Moreover, the local-global network (LG-Net) is designed to model local and global information. Specifically, the self-attention layer [18] is stacked to the 1D temporal convolution layer to capture the long-term and short-term temporal information. Experimental results demonstrate that the proposed text anchor based metric learning method shows consistent improvements over speech anchor on representative CNN-based models. Besides, our LG-Net model achieves SOTA accuracy of 97.67% and 96.79% on Google Speech Commands Dataset version 1 (GSCDv1) and version 2 (GSCDv2), respectively.

2. Proposed Method

2.1. Data Process

The raw audio is decomposed into a sequence of frames where the window length is 25 ms and the stride is 10 ms for feature extraction. We use 40 Mel-Frequency Cepstral Coeffi-

*Corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-136

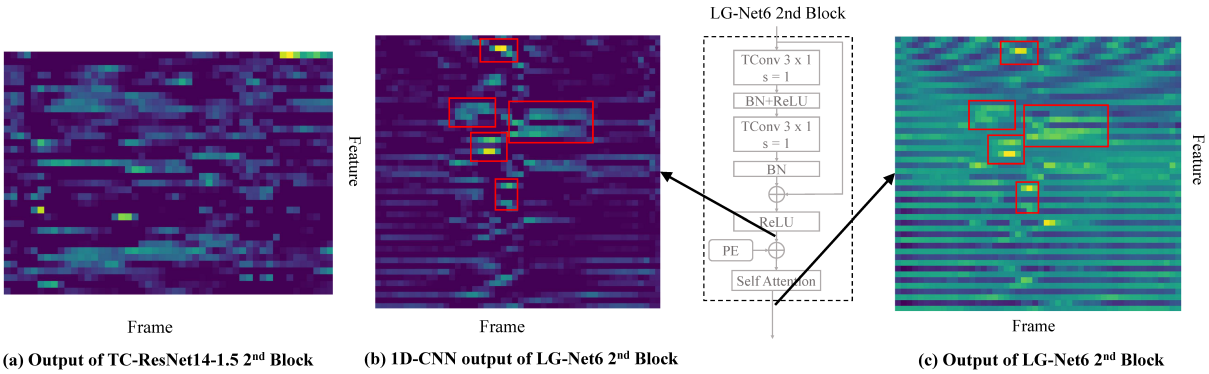


Figure 1: *Feature maps produced by the TC-ResNet14-1.5 and proposed model LG-Net using a benchmark image, saliency for each feature is represented by brightness. (a) and (b) shows that the feature maps outputs by 1D-CNN are sparse, representing that 1D-CNN focuses on short-term temporal information. (c) shows the self-attention layer modeling the long-term information while preserving the short-term temporal information captured by the 1D-CNN.*

cient (MFCC) features for each frame and stack them over the time-axis, denote as $M \in \mathbb{R}^{T \times F}$ where F represents the dimension of the MFCC feature, and T denotes the number of frames. The text anchor is extracted by BERT-base [17] denoted as $V \in \mathbb{R}^{D}$, where D represents the dimension of the text anchor (*i.e.* $D = 768$).

2.2. Text anchor based metric learning method

As mentioned in Section 1, the susceptibility of the speech anchor leads to the poor performance of the model. Therefore, we propose a text anchor based metric learning method for KWS, illustrated in Figure 2b. The input triplet is denoted as (M^+, M^-, V), where M^+ and M^- are input MFCCs from different classes, and V as the input text vector corresponding to M^+. The speech embeddings $E_{\mathrm{S}}^{M^+} \in \mathbb{R}^{D'}$ and $E_{\mathrm{S}}^{M^-} \in \mathbb{R}^{D'}$ are extracted from M^+ and M^- by the speech embedding extraction module f_{S}, respectively, where D' represents the dimension of the speech embedding (*i.e.* $D' = 128$). In parallel, V is mapped as a text embedding $E_{\mathrm{T}} \in \mathbb{R}^{D'}$. The triplet loss is employed to decreases the distance between the E_{T} and $E_{\mathrm{S}}^{M^+}$ and increases the distance between E_{T} and $E_{\mathrm{S}}^{M^-}$. For a single sample, the triplet loss is thus

$$L_{tri} = \max\left(d\left(E_{\mathrm{S}}^{M^+}, E_{\mathrm{T}}\right) - d\left(E_{\mathrm{S}}^{M^-}, E_{\mathrm{T}}\right) + \alpha, 0\right) \quad (1)$$

where $d(x, y) = \|x - y\|_p$ is the pairwise-distance between x and y (*i.e.* $p = 2$); α is a constant margin (*i.e.* α = 1).

Note that the phone embeddings extracted by the text encoder in [7] is fundamentally different from the text anchor proposed in this paper. In [7], a pronunciation dictionary is used to map words to their sequence of phonemes. Therefore, the phone embeddings do not contain the information about other words, although it is not influenced by factors such as acoustic environment and speaker. The text anchor in this paper is extracted from the BERT [17] which is trained on the large-scale text corpus, it contains information about other words.

2.3. Local-global Neural Network

As mentioned in Section 1, for KWS, long-term temporal information is crucial because the characteristics of keywords are usually different on the time scale. In this work, we utilize the

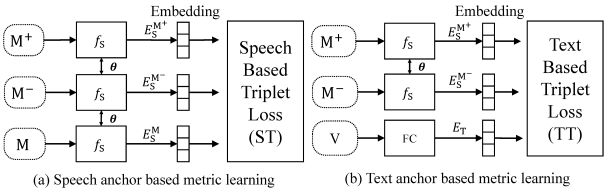


Figure 2: *(a) Speech anchor based metric learning and (b) proposed text anchor based metric learning. M^+ and M^- are input MFCCs from different classes, M is input MFCC from the same class of M^+, and V as the input text vector encoded by BERT [17] corresponding M^+. E_{S} represents the speech embedding output from the speech embedding extraction module f_{S}. E_{T} represents the text embedding.*

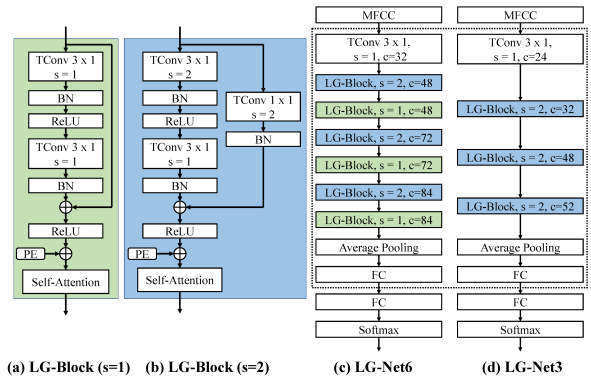
self-attention [18] layer to capture long-term temporal information. The temporal convolution with residual structure [13] is adopt in this paper, which is one of the most widely used 1D-CNN architectures for KWS. To reduce the number of parameters, 3×1 kernels are utilized instead of 9×1 kernels. As Figure 3 shows, and the self-attention layer is stacked to the 1D-CNN layers, denoted as LG-Block. Following the design of the previous study [13], the identity shortcuts can be directly used when the input and output have matching dimensions (Figure 3a); otherwise, we use an extra 1×1 convolution with a stride to match the dimensions (Figure 3b). Local-global neural network (LG-Net) can be easily constructed by stacking LG-Blocks. We select LG-Net6 (Figure 3c), which has 6 LG-Blocks as our base model. To establish a small-footprint model, we compress the LG-Net6 by taking out three LG-Blocks and reducing the number of channels, named LG-Net3.

Considering the variation in feature maps due to speaking style, speed of speech, we use an averaging pooling strategy on the time dimension. The output of the average pooling layer is given to a fully connected layer to generate speech embedding $E_{\mathrm{S}} \in \mathbb{R}^{D'}$.

$$E_{\mathrm{S}} = f_{\mathrm{S}}\left(M; \theta_{\mathrm{S}}\right) \quad (2)$$

where θ_{S} denote the parameters of the f_{S}.

Finally, the predicted score, $\hat{y} \in \mathbb{R}^{C}$ where C denote the



(a) LG-Block (s=1) (b) LG-Block (s=2) (c) LG-Net6 (d) LG-Net3

Figure 3: *The local-global block (LG-Block) when (a) stride = 1 and (b) stride = 2 and two implementations of local-global neural network (LG-Net) when (c) LG-Net6 and (d) LG-Net3. TConv means the temporal convolution layer, "s" and "c" indicates stride and channel of TConv, respectively. PE means the position encoding [18]. BN and FC denote batch normalization and fully connected layer. The part inside the dotted line is regarded as the speech embedding extraction module.*

number of class, is obtained from E_{S}

$$\hat{y} = \sigma\left(f_{\mathrm{FC}}\left(E_{\mathrm{S}}; \theta_{\mathrm{FC}}\right)\right) \tag{3}$$

where σ is the sigmoid function; θ_{FC} denotes the parameters of the fully connected layer f_{FC}. KWS as a multi-classification problem, the cross entropy (CE) loss is adopted to optimize the model, for the given ground truth $y \in \mathbb{R}^C$ (where $y^{(i)} = \{0, 1\}$ denotes whether label i appears or not), the loss L is calculated using binary cross-entropy:

$$L_{CE} = \sum_{i=1}^{C} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right) \tag{4}$$

In this work, the networks are optimized by the loss L, which is calculated by weighted sum of L_{tri} and L_{CE}

$$L = \beta L_{\mathrm{tri}} + (1 - \beta) L_{\mathrm{CE}} \tag{5}$$

where β is the balanced weight between L_{tri} and L_{CE}.

Except for the cases of target keywords and non-target words, a "silence" class is considered to indicate the case when no speech is detected [19]. However, in this work, the text vector of "silence" does not match the definition in KWS, so the model cannot be trained directly using the text vector of "silence". To address the problem, we first train the network without the "silence" sample and then finetune the two FC layers on the full dataset with only CE loss.

3. Experiments

3.1. Datasets

The proposed models are trained and evaluated with Google Speech Commands Dataset version 1 (GSCDv1[1]) and version 2 (GSCDv2[2]) [19]. Following the implementation of Google [19], we seek to discriminate among 12 classes for GSCDv1: "yes", "no", "up", "down", "left", "right", "on", "off", "stop", "go", unknown, or silence, and 16 classes for GSCDv2 which add 4 keywords of GSCDv1: 'backward", "forward", "follow", and "learn". In order to generate background noise, we randomly sample and crop background noises provided in the dataset. For a fair comparison, in our test set, the "silence" class test samples are taken from open source speech commands dataset test set version 2[3] [19], and test samples of other classes are written in the officially released *testing.list*[12]. Table 1 shows the number of samples in training, validation, and test sets. Note that to make it easy for other researchers to reproduce the model and results in this paper, we do not use any data augmentation methods.

Table 1: *The number of samples in training, validation and test set of google speech commands dataset version 1 and version 2.*

Dataset	Train	Valid	Test	Total
GSCDv1	55287	7218	7242	69747
GSCDv2	89043	10401	11413	110857

3.2. Implementation Details

Our implementation was done with Pytorch [20] deep learning toolkit.

Training. All the models are trained with a mini-batch of 256 samples using stochastic gradient descent with weight decay of 0.001 and momentum of 0.9. For formula (5), the L_{tri} and L_{CE} weights to be the same, *i.e.* $\beta = 0.5$, $L = 0.5L_{\mathrm{CE}} + 0.5L_{\mathrm{tri}}$. The initial learning rate is set to be 0.01 and decayed by a factor of 3 when the validation accuracy does not increase for 3 epochs. The training is terminated when validation accuracy does not increase for 10 epochs. For the finetuning stage mentioned in Section 2.3, the training setup is the same as the text anchor based metric learning method, except that β is set as 0.

Evaluation. The primary metric in our experiments is classification accuracy, the proportion of correct decisions out of the total number of samples. We also report the number of parameters and the false reject rate (FRR) at 0.5% false alarm rate (FAR). We train each model 10 times and report its average performance.

3.3. Baselines

To demonstrate the consistent performance improvement of our proposed text anchor based metric learning method, we reproduce CnnTradFpool3 [9], CnnOneFstride4 [9], TDNNSWSA [14], TC-ResNet8 [13], and TC-ResNet14-1.5 [13], which are representative CNN-based KWS models.

We use three loss functions for training and comparison. First, the CE loss, which is the most widely used loss function in the KWS task. Second, the triplet loss based on speech anchor (*ST*) is used to evaluate the performance gains from the triplet loss. As shown in Figure 2a, the input triplet is denoted as (M^+, M^-, M), where M is input MFCC randomly sampled from the training set with the same class of M^+. Third, the triplet loss based on text anchor (*TT*). As mentioned in Section 2.3, we employ CE loss accompanied by the ST and the TT, denote as *CE+ST* and *CE+TT*, respectively. Also, to evaluate the performance improvement brought by our proposed LG-Net model, we selected several CNN-based KWS models trained with CE loss functions for direct comparison [10, 15, 16].

[1] http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz
[2] http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz
[3] http://download.tensorflow.org/data/speech_commands_test_set_v0.02.tar.gz

Table 2: *Comparison of the proposed method and the baselines*

Model	Loss	Params	Acc.(v1)	Acc.(v2)
CnnTradFpool3 [9] (Sainath *et al.* 2015)	CE	1.39M	93.53%	90.15%
	CE+ST		93.72%	90.19%
	CE+TT		94.35%	91.45%
CnnOneFstride4 [9] (Sainath *et al.* 2015)	CE	3.84M	91.50%	89.16%
	CE+ST		91.68%	90.03%
	CE+TT		92.21%	90.38%
TDNNSWSA [14] (Bai *et al.* 2019)	CE	20K	94.19%	93.11%
	CE+ST		95.32%	93.77%
	CE+TT		95.47%	94.15%
TC-ResNet8 [13] (Choi *et al.* 2019)	CE	72K	95.71%	95.22%
	CE+ST		96.12%	95.56%
	CE+TT		96.58%	95.59%
TC-ResNet14-1.5 [13] (Choi *et al.* 2019)	CE	313K	96.78%	95.96%
	CE+ST		96.81%	96.20%
	CE+TT		96.89%	**96.27%**
Res15* [10] (Tang *et al.* 2018)	CE	238K	95.80%	-
TENet12* [15] (Li *et al.* 2020)	CE	100K	96.84%	-
NAS2* [16] (Mo *et al.* 2020)	CE	886K	**97.22%**	-
LG-Net3 (ours)	CE	74K	95.71%	95.35%
	CE+ST		96.46%	95.39%
	CE+TT		96.82%	95.77%
LG-Net6 (ours)	CE	313K	97.03%	96.45%
	CE+ST		97.13%	96.57%
	CE+TT		**97.67%**	**96.79%**

* represents the direct result of the corresponding paper.

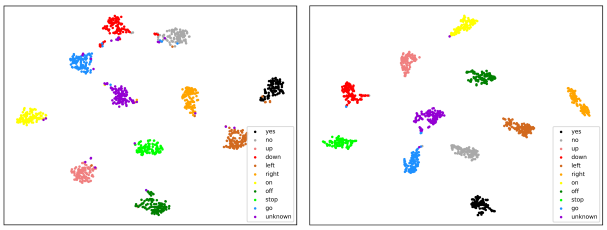
Table 3: *Comparison of FRR (false reject rate) of TC-ResNet14-1.5 and LG-Net6, FRR (false alarm rate) is at 0.5%*

Model	Params	Loss	FRR
TC-ResNet14-1.5	313K	CE	5.91%
		CE+ST	5.36%
		CE+TT	5.29%
LG-Net6 (ours)	313K	CE	4.69%
		CE+ST	4.31%
		CE+TT	**3.56%**

3.4. Experimental Results and Analysis

Impact of text anchor based metric learning method. As Table 2 shows, for each model, training with *CE+TT* loss shows consistent improvements over training with *CE+ST* loss, since that BERT provides informative and robust text anchors, which alleviates the effects of the acoustic environment and speakers. Furthermore, We find that training with *CE+ST* loss yields a performance gain compared to CE loss because triplet loss increases the distinguishability of features. From Table 3, we can see that LG-Net6 achieves a lower FRR at 0.5% FAR. Note that there is no increase in memory footprint and computational resource at inference compared with the corresponding baseline. As Figure 4 shows, we randomly sample speech embeddings and use the t-SNE algorithm [23] to visualize the embeddings. We can see that keywords with similar pronunciation are difficult to distinguish (e.g. "go" and "no"), but are easily distinguished in the feature space after using our proposed text anchor based metric learning method.

Impact of other text vectors. We make a step forward to study how other text vectors influence the performance of our proposed method. Comparison experiments among the different layers of BERT and Glove [21] are performed. Referring to Table 4, both models, TC-ResNet14-1.5 [13] and LG-Net6, achieve the best performances by employing the output of the first layer from BERT as the text anchor. The results indicate that the lower layer of BERT can mostly capture word-level information, which is consistent with the statement in [22].



(a) Trained by CE loss (b) Trained by proposed text based triplet loss

Figure 4: *t-SNE visualization of speech embeddings extracted by LG-Net6.*

Table 4: *Comparison of other text embeddings on GSCDv2*

Model	Text Embedding	Acc.
TC-ResNet14-1.5 [13] (Choi *et al.* 2019)	glove.840B.300d [21]	96.21%
	BERT-Layer 1 [17]	**96.27%**
	BERT-Layer 2 [17]	96.23%
	BERT-Layer 12 [17]	96.22%
LG-Net6 (ours)	glove.840B.300d [21]	96.47%
	BERT-Layer 1 [17]	**96.79%**
	BERT-Layer 2 [17]	96.52%
	BERT-Layer 12 [17]	96.50%

Impact of the local-global block. As shown in Table 2, compared to models trained using CE loss, LG-Nets achieve a balance between model size and accuracy. Compared to TC-ResNet14-1.5 [13], our LG-Net6 achieves better accuracy of 97.03% and 96.45% on GSCDv1 and GSCDv2, respectively. Compared to NAS2 [9], our proposed LG-Net6 achieves comparable results using only 35% (313K/886K) of the number of parameters. The small-footprint model LG-Net3 obtains 95.71% KWS accuracy in GSCDv1 and 95.35% KWS accuracy in GSCDv2. Visualization results (Figure 1b and Figure 1c) show that the self-attention layer helps the network capture long-term temporal information while preserving the short-term temporal information captured by the 1D-CNN.

4. Conclusion

In this paper, we propose text anchor based metric learning method and design a neural network LG-Net for small-footprint keyword spotting (KWS). The experimental results show that the proposed text anchor based metric learning method shows consistent improvements over the speech anchor based method. Moreover, our LG-Net model achieves SOTA accuracy on the Google Speech Commands Dataset version 1 and version 2.

5. Acknowledgements

This paper was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: JSGG20191129105421211 & GXWD20201231165807007-20200814115301001)

6. References

[1] J. Huh, M. Lee, H. Heo, S. Mun, and J. S. Chung, "Metric learning for keyword spotting," *arXiv preprint arXiv:2005.08776*, 2020.

[2] A. Raju, S. Panchapagesan, X. Liu, A. Mandal, and N. Strom, "Data augmentation for robust keyword spotting under playback interference," *arXiv preprint arXiv:1808.00563*, 2018.

[3] B. Liu, S. Nie, Y. Zhang, S. Liang, Z. Yang, and W. Liu, "Loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6361–6365.

[4] K. Zhang, Z. Wu, D. Yuan, J. Luan, J. Jia, H. Meng, and B. Song, "Re-weighted interval loss for handling data imbalance problem of end-to-end keyword spotting," *Proc. Interspeech 2020*, pp. 2567–2571, 2020.

[5] 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 *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4704–4708.

[6] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Mining effective negative training samples for keyword spotting," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7444–7448.

[7] N. Sacchi, A. Nanchen, M. Jaggi, and M. Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *INTERSPEECH 2019-IEEE International Conference on Acoustics, Speech, and Signal Processing*, no. CONF, 2019.

[8] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[9] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[10] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5484–5488.

[11] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," *arXiv preprint arXiv:1803.10916*, 2018.

[12] H. Zhang, J. Zhang, and Y. Wang, "Sequence-to-sequence models for small-footprint keyword spotting," *arXiv preprint arXiv:1811.00348*, 2018.

[13] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," *arXiv preprint arXiv:1904.03814*, 2019.

[14] Y. Bai, J. Yi, J. Tao, Z. Wen, Z. Tian, C. Zhao, and C. Fan, "A time delay neural network with shared weight self-attention for small-footprint keyword spotting," *Proc. Interspeech 2019*, pp. 2190–2194, 2019.

[15] X. Li, X. Wei, and X. Qin, "Small-footprint keyword spotting with multi-scale temporal convolution," *arXiv preprint arXiv:2010.09960*, 2020.

[16] T. Mo, Y. Yu, M. Salameh, D. Niu, and S. Jui, "Neural architecture search for keyword spotting," *arXiv preprint arXiv:2009.00165*, 2020.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[22] G. Jawahar, B. Sagot, and D. Seddah, "What does bert learn about the structure of language?" in *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.